Learning to Generate and Refine Object Proposals

Haoyang Zhang

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

May 2018

Except where otherwise indicated, this thesis is my own original work.

Haoyang Zhang
6 May 2018

Acknowledgments

First of all, I would like to thank my primary supervisor, Dr. Xuming He. I specially thank Xuming for introducing me to the research world. Xuming is an excellent researcher with solid background in computer vision and machine learning. He is conscientious, responsible and cooperative, often giving me valuable detailed advice on my research. I learnt a great deal from Xuming, including the research methods, experiment design, paper reading and writing skills, idea demonstration and a body of knowledge in computer vision *etc*. Without his invaluable help and support, it is impossible for me to complete my PhD.

I am also grateful for my other supervisory panel members: Lexing Xie, Miaomiao Liu, Laurent Kneip and Fatih Porikli. Thank you very much for the support and discussion. A special thanks goes to Lexing, who brought me the opportunity of studying in Australia in the beginning and helped me finish my PhD in the end.

I would also like to thank my friends at Data61/NICTA and ANU: Weipeng Xu, Buyu Liu, Wei Zhuo, Fang Wang, Hongtao Yang, Hao Zhou, Wenbing Huang and Tong Zhang. Thank you for helping me in my life, sharing joys with me and supporting me. Weipeng gave me a lot of help when I arrived in Australia and shared me with many interesting things. Hao, Wenbing and Tong are fun to play with and we enjoyed many memorable dinners, which helped me go through those tough days.

I also acknowledge the Australian National University (ANU), and Data61 / National Information and Communication Technology Australia (NICTA), Commonwealth Scientific and Industrial Research Organisation (CSIRO), for providing me scholarships to undertake the research for my PhD studies.

Finally, I thank my family members for all the love and understanding.

Abstract

Visual object recognition is a fundamental and challenging problem in computer vision. To build a practical recognition system, one is first confronted with high computation complexity due to an enormous search space from an image, which is caused by large variations in object appearance, pose and mutual occlusion, as well as other environmental factors. To reduce the search complexity, a moderate set of image regions that are likely to contain an object, regardless of its category, are usually first generated in modern object recognition subsystems. These possible object regions are called object proposals, object hypotheses or object candidates, which can be used for down-stream classification or global reasoning in many different vision tasks like object detection, segmentation and tracking, *etc*.

This thesis addresses the problem of object proposal generation, including bounding box and segment proposal generation, in real-world scenarios. In particular, we investigate the representation learning in object proposal generation with 3D cues and contextual information, aiming to propose higher-quality object candidates which have higher object recall, better boundary coverage and lower number. We focus on three main issues: 1)how can we incorporate additional geometric and high-level semantic context information into the proposal generation for stereo images? 2) how do we generate object segment proposals for stereo images with learning representations and learning grouping process? and 3) how can we learn a context-driven representation to refine segment proposals efficiently?

In this thesis, we propose a series of solutions to address each of the raised problems. We first propose a semantic context and depth-aware object proposal generation method. We design a set of new cues to encode the objectness, and then train an efficient random forest classifier to re-rank the initial proposals and linear regressors to fine-tune their locations. Next, we extend the task to the segment proposal generation in the same setting and develop a learning-based segment proposal generation method for stereo images. Our method makes use of learned deep features and designed geometric features to represent a region and learns a similarity network to guide the superpixel grouping process. We also learn a ranking network to predict the objectness score for each segment proposal. To address the third problem, we take a transformation-based approach to improve the quality of a given segment candidate pool based on context information. We propose an efficient deep network that learns affine transformations to warp an initial object mask towards nearby object region, based on a novel feature pooling strategy. Finally, we extend our affine warping approach to address the object-mask alignment problem and particularly the problem of refining a set of segment proposals. We design an end-to-end deep spatial transformer network that learns free-form deformations (FFDs) to non-rigidly warp the shape mask towards the ground truth, based on a multi-level dual mask

feature pooling strategy. We evaluate all our approaches on several publicly available object recognition datasets and show superior performance.

Contents

List of Figures

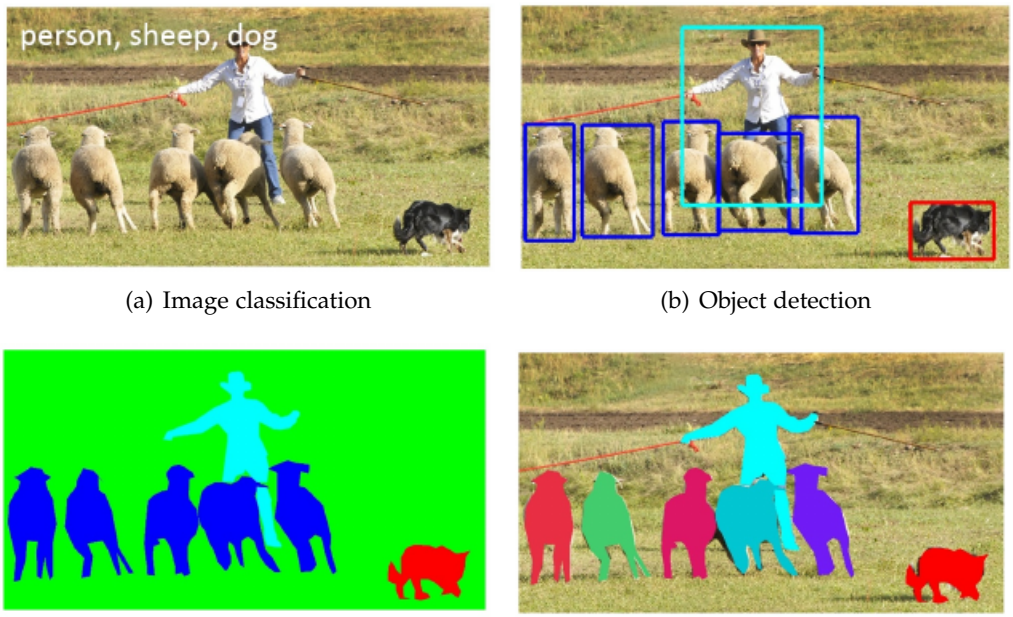
List of Tables

Chapter 1

Introduction

1.1 Overview

Object recognition is a core function of human vision system and also a fundamental task in computer vision. By simply looking at an image, we can easily separate all the objects present on it and recognize them effortlessly. To endow a computer with such functions, researchers have made great endeavour in the past fifty years. Despite the exciting progress that the community has made, however, it is still particularly difficult and challenging to build such a computer system to achieve human-level performance and efficiency in a generic environment.



(a) Image classification

(b) Object detection

(c) Semantic segmentation

(d) Object instance segmentation

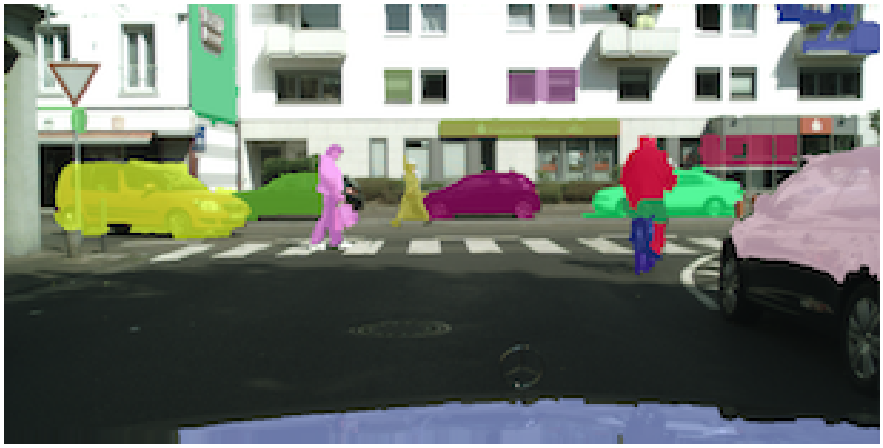
Figure 1.1: Illustration of four different levels of object recognition in computer vision: (a) image classification, (b) object detection, (c) semantic segmentation and (d) object instance segmentation. Image taken from [1].

The main difficulty of object recognition lies in the large variations of visual objects, including object appearance, viewpoint, size, pose, occlusion and illumination. This leads to a vast search space to find a match between image evidence and object models. To design an algorithm to handle all these situations, we have to confront with the serious problem of high computational complexity. Take object detection (see Figure 1.1(b)) for example. As one of the sub-tasks of object recognition, object detection aims to localize each of target objects in an image with a bounding box and assign a class label to each box. Until recently, the most successful object detection algorithms employ the well-known "sliding window" paradigm, in which object detectors are applied at every object location and scale of an image [19, 20, 2] to detect all different target objects. In general, it requires to test $10^6 \sim 10^7$ windows per image for multi-scale detection. With the ever increasing complexity of the classifiers, the computation cost of the detection algorithms grows significantly and applying the complex classifiers to millions of windows becomes extremely inefficient.

By contrast, humans can rapidly detect and recognize objects. One reason is that human vision system works with an efficient mechanism. Instead of processing a whole scene at once, humans tend to first focus attention on a selected subset of parts in the visual space and then move the fovea to other related parts to build up an internal interpretation of the whole scene [21, 22, 23]. This visual attention mechanism significantly decreases the task complexity as it puts the resources of visual processing only on the objects of interest and ignores those irrelevant visual clutters. At the same time, evidence from cognitive psychology [24, 25] and neurobiology [26, 27] also shows that humans visual system tends to pay attention to possible object locations before reasoning them in detail. This efficient mechanism in human visual perception suggests that before identifying the objects in an image it is reasonable to produce a small set of possible objects, that is, object proposals, object hypotheses or object candidates.

Generally speaking, object proposal is a region (a bounding box or a segment) in an image which is likely to contain an object, regardless of its category. (see Figure 1.2 for some examples.) Actually back to 1970s, researchers had studied the problem of grouping a given point set into different subsets, *i.e.* object proposals, for detection through graph-theoretical algorithms [28, 29], based on the Gestalt principles described by Wertheimer [30]. Modern visual object localization and recognition systems have also adopted a similar strategy as in the human attention system, which commonly generate a set of object proposals in the first stage to address the efficiency issue [4, 31, 32, 10, 3]. The main idea is to fast generate a small set of object proposals before classification for a given image. If this small set of object proposals can achieve high object recall, the detection accuracy will be kept but the efficiency will be significantly improved, because in general only hundreds of or at most thousands of object proposals are generated for one image, which is significantly smaller than the sliding widows. Besides the greatly increased efficiency, by focusing on a relatively small set of object-like regions, it enables us to use better object representations and improve significantly the accuracy of target vision tasks. In addition, thanks to much fewer background clutters in the training examples, the imbalance between

(a) Bounding box proposals



(b) Segment proposals

Figure 1.2: Illustration of object proposals: (a) bounding box proposals and (b) segment proposals.

the positive and negative examples is considerably lower than that in the traditional sliding-window methods , and with a relatively smaller training set, the classifier is able to focus on the class boundary region. Therefore, the training of classifiers turns to be more easy and effective, boosting the system performance further [8].

The initial success of the bounding box proposal in object detection has inspired the research of generating the segment proposal, which pushes the boundaries of object proposal research further. Compared with bounding box proposals, segment proposals take the form of segmentation masks and seek to identify the spatial extent of objects, so they are more informative but more challenging to generate. Segment proposals are usually used in the task of object instance segmentation (see Figure 1.1(d)) whose goal is to not only detect all objects in an image but also segment each object instance, which is hence more difficult. The use of segment proposals makes the pipeline of instance segmentation share the similar paradigm of object detection. In this way, the advance in object detection can be easily applied to instance segmentation after minor modifications. This brings recent rapid progress in object instance segmentation.

These advantages have attracted much attention from the community and significant progress has been made in the area of object proposal generation. At present,

object proposal has played a vital role in many vision tasks, such as object detection [32, 8, 10], object instance segmentation [33, 3, 9] and visual object tracking [34, 35], which has far exceeded its original application scope. In general, an object proposal generation method should have the following properties: (1) high recall, ideally all target objects should be recalled by the generated object proposals, (2) low count, the high recall should be obtained using a relatively modest number of object proposals, (3) good precision, proposed object regions, regardless of bounding boxes or segment masks, should match the object as accurately as possible and (4) computational efficiency.

Recent years, while the progresses in object proposal generation have improved the performance of object recognition tasks, there are still two serious limitations in the prior proposal methods. First, unlike the human visual system, most object proposal approaches focus on 2D images and are unable to make use of multi-modal cues, *e.g*. the depth cue, as well as additional semantic information. In addition, in contrast to the bounding box candidate generation, generating accurate object segment proposals is still an extremely challenging task due to the large variations in the object boundaries. Especially when the number of proposals is small and the IoU threshold is high, the segment proposals usually present very low recall rates. Since the recall rate of object proposals imposes an upper bound for the instance segmentation accuracy, this obviously has an adverse impact on this task that most unrecalled object cannot be detected.

In this thesis, we focus on developing algorithms to produce high-quality object proposals, especially the segment proposals. First, we aim to extend the object proposal generation to stereo images, which has not been extensively investigated. We would like to explore the geometric information provided by stereo images as well as semantic context from scene labelling in object proposal generation. Our research question is whether these two types of high-level information can help improve the recall rate of object proposals and at the same time reduce the number of proposals. Second, we intend to introduce the representation learning and similarity learning into the segment proposal generation. We will investigate if the features and the grouping strategy learned by deep networks can benefit segment proposal generation. Finally, we target at improving the precision of object proposals through warping based on contextual information. We will study the method to refine either bounding box proposals or segment proposals by changing their locations or their shapes so that they become closer to their ground truth.

In the following section, we will discuss the main issues of previous methods in detail. Section 1.3 outlines the main ideas of our solutions to the main issues. Section 1.4 briefly introduce the content of each chapter, and Section 1.5 summarizes the main contributions of this thesis.

1.2 Main Issues in Object Proposal Generation

We first summarize the prior approaches to generating bounding box candidates and segment proposals, which will facilitate our discussion below.

To generate bounding box proposals, the widely adopted strategy is a window-scoring method. It generates object proposals by scoring each candidate window according to how likely it is to contain an object, that is, assigning an objectness score (the probability of containing an object) to each candidate window. After that, certain number of them are chosen as object proposals by using a threshold. In order to perform efficiently, such a method usually employs a simple classifier and simple features to compute the objectness score for each candidate window [4, 36, 37]. In the early stage, this method generally utilizes multiple hand-crafted low-level cues such as colour, edge, location, size and gradient to calculate the objectness score. With the prevalence of convolutional neural networks (CNNs) at present, more powerful learned deep features are used to train a predictor of objectness score for candidate windows [38, 39]. Alternatively, the bounding box proposal can also be obtained by taking the tight box enclosing a segment proposal.

Compared to producing bounding box proposals, generating object segment proposals entails both object-level localization and pixelwise perceptual grouping, which is more challenging. Given an image, the space of all possible segments that can be obtained is extremely large, which means a sliding-segment approach that selects a specified shape of segment to slide across the image is not realistic. So most early works usually rely on perceptual grouping methods that attempt to generate multiple segments which are likely to correspond to objects. These methods can be separated into two main categories:(1) superpixel merging [40, 5], where multiple over-segmentations are merged into region proposals according to various heuristics and (2) seed segmentation [31, 41, 42], where multiple segmentation problems with diverse seeds are solved to generate a foreground-background segmentation for each seed and these segmentation masks are taken as segment proposals. More recent approaches use deep convolutional networks to learn the feature representation and directly predict class-agnostic object masks [3, 11] from image patches, which obtain impressive performance. Most of segment proposal methods also learn a regressor to estimate how likely a segment cover an object, *i.e*. output an objectness score for each segment proposal.

In this thesis, we aim to address several main limitations in the existing object proposal generation methods, and propose the following three research problems in order to handle wider scope of real-world scenarios and further improve the proposal qualities.

1) Generating object proposals with geometric features and semantic context. Despite the rapid progress in object proposal generation, few methods have considered generating object proposals with high-level geometric features and semantic context for stereo images. Most of existing approaches generate proposals from single modality and focus on low- or mid-level features. These methods mainly work

in the RGB space and extract color, edge and size *etc*. features to compute objectness scores. They rarely investigate the setting of stereo images and exploit semantic contextual information. The low-level intensity-based information can be an indicator of an object to certain extent, but it can be insufficient and ambiguous in challenging scenarios. On the other hand, the spatial locations of object instances need to satisfy certain geometric/physical constraints and have close relations to their neighbouring object classes, such as supporting relation and relative size. Take cars for example, although they varies in appearance, cars generally have a typical size and tend to occur on the road. Hence, we argue that geometric and semantic context cues can benefit the proposal generation and further improve their quality. Additionally, high-level abstract semantic representation learned by deep neural networks is a strong discriminative cue to separate the object from the background. Therefore, integrating deep features into the framework of object proposal generation is also able to boost its performance. These arguments bring the first problem: how do we incorporate additional geometric and high-level semantic context information into the proposal generation for stereo images?

2) Generating object segment proposals for stereo images with learning representations and learning grouping process. Most object segment proposal methods which adopt the superpixel grouping pipeline decide whether merge two adjacent superpixels or not according to the edge strength or the similarity between these two superpixels. They usually represent superpixels using low-level image features like size, location, shape and color, and based on these cues they compute the edge strength or the similarity between adjacent superpixels. These features only capture the local properties of the superpixel and fail to encode the global and context information of the image, which probably bring inaccuracy into the superpixel merging process. On the other hand, convolutional neural networks (CNN) show great power in feature representation learning. Its multiple layers naturally represent different level of features. The earlier layers represent simple aspects of the image, such as edges and colors, while the latter layers describe increasingly sophisticated aspects of the image, such as shapes and patterns. This makes the multiple level features learned by CNN quite suitable to represent a superpixel. Moreover, geometric features are informative in separating objects from the background, but most of segment proposal methods rarely incorporate geometric features into their framework. Therefore, in order to generate better segment proposals through superpixel merging for stereo images, we first need to solve these problems: how to learn feature representations for superpixels with the CNN? and what geometric features need to extract for superpixels?

Assuming we manage to encode the superpixel with CNN features and geometric features, we also need to consider the process of merging superpixels. Most existing methods manually design the similarity or dissimilarity metrics between superpixels and rely on the designed metrics to iteratively merge superpixels. As the learned CNN features are much more complex than low-level hand-crafted features and at the same time the geometric features are quite different from appearance features, it

would be very difficult to design effective superpixel similarity metrics in this setting. Further, since similar superpixels may come from different object instances, we need an effective mechanism to make sure not to merge them in the grouping process. To overcome these difficulties, we would ask: Can we design a network to learn such a grouping process?

3) Learning to refine object segment proposals. More recent approaches to generating object segment proposals learn deep networks to produce binary masks from the image directly, showing impressive performance. Nevertheless, learning such a direct mapping from images to segments has shown to be challenging, which usually produces object masks lacking good boundary alignment and requires post-processing to improve their quality. To generate better object proposals, an alternative approach is to refine an initial set of object proposals produced by existing methods through warping them to improve their alignment precision. Such a strategy enables us to use the initial proposal as a starting point and learn additional feature representations for improving their qualities. Hence it is more flexible. In addition, as it aims to minimize the residual error between the initial proposals and the ground truth, the problem of refinement is conceptually simpler than solving the original task, especially for generating segment proposals. However, only a few attempts have been made to improve the quality of initial segment proposals. One possible reason is that, unlike bounding boxes, segments have arbitrary shapes and hence it is difficult to model the deformations between the segment proposal and its groundtruth mask. Thus, to implement the idea of refining the segment proposals through warping, we need to solve the following problems: how can we model the deformations between the object segment and its groundtruth mask? Can we design deep networks to predict the underlying spatial transforms, including affine transformations and free-form deformations in this context?

These aforementioned issues and the solutions provided constitute the center of this thesis. The following section will provide an overview of our solutions to these problems.

1.3 Our Methods

To solve the problems discussed above, we propose a series of approaches based on multi-cue fusion and representation learning. In what follows, we will give an overview of each approach and briefly describe the modules of each method.

1) Semantic context and depth-aware object proposal generation. To benefit from geometric features provided by stereo images and to leverage the high-level semantic contextual information, we propose a semantic context and depth-aware object proposal generation method. In this work, we take a pair images from a stereo camera as input and start from a set of initial object proposals generated by an existing method. Our goal is to refine this set of proposals by re-ranking them and fine-tuning their spatial locations based on a new set of object and context informa-

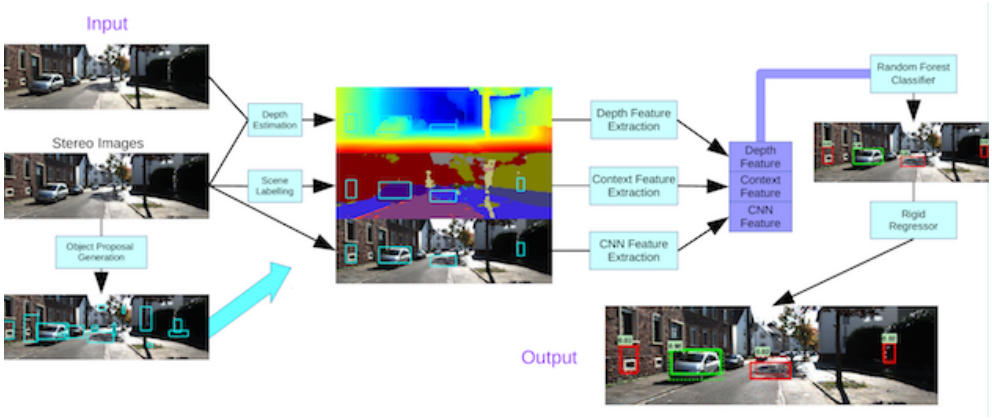


Figure 1.3: Overview of our approach for semantic context and depth-aware object proposal generation. The input are a pair of stereo images and an initial set of proposals. We extract three types of object and context cues, and use them to re-rank the proposals and refine their locations.

tion.

Concretely, we consider the following three kinds of objectness cues. First, we use the noisy depth computed from the stero images to estimate a set of geometric features on each object candidate; second, we design a semantic context feature to describe the surrounding object class distribution, which is computed from a noisy semantic labelling; finally, we extract a CNN feature from each object candidate. We then fuse these object and context cues to re-rank and relocate the initial object candidates. In particular, based on those features, we train a classifier to predict a new objectness score for each candidate, and regressors to adjust the location of its bounding box. An overview of our method is illustrated in Figure 1.3.

2) Learning to generate object segment proposals with multi-modal cues. To make use of learned deep features and geometric information, we propose a learning-based object segment proposal generation method for stereo images. We take an alternative deep learning approach to efficiently incorporate the depth cues computed from the stereo, and learn an iterative merging process for generating a diverse set of high-quality region proposals. Unlike the previous global approaches that learn an image-to-mask mapping, we mainly focus on learning a representation for object-driven perceptual grouping, which is an easier problem due to its local nature and potential to be modelled by a simpler network. More importantly, it enables us to design a late fusion strategy to incorporate the noisy depth cues into grouping without retraining the full deep network pipeline.

Specifically, our method consists of two stages. We start from an initial segmentation hierarchy of the left image of the stereo images and sequentially merge neighboring regions in each level of the hierarchy based on affinity scores predicted by a learned similarity network. This merging process generates new hierarchies of image segments, which are used to produce a pool of regional proposals by tak-

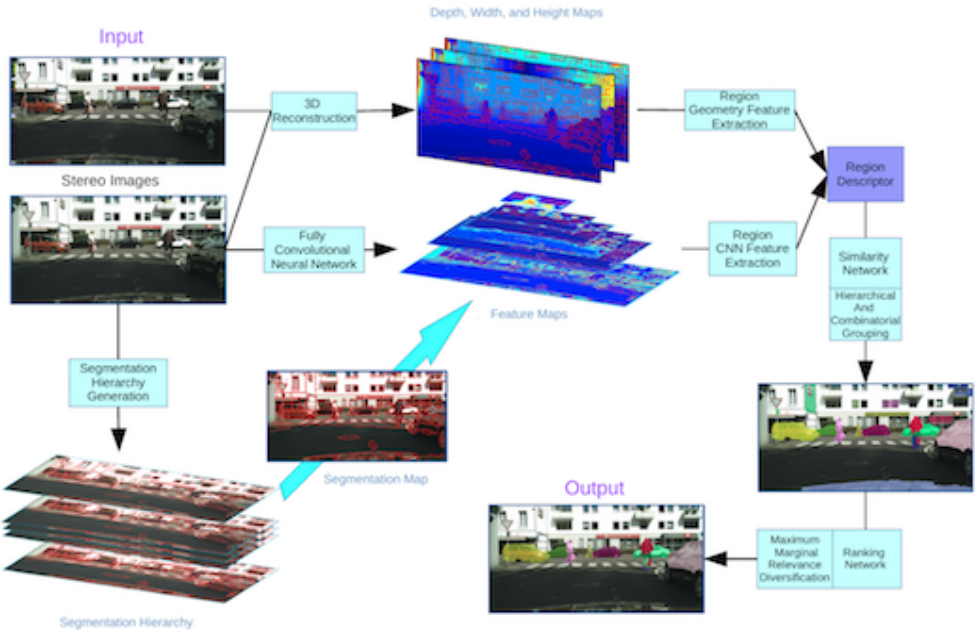


Figure 1.4: Illustration of our system for learning to generate object segment proposals with multi-modal cues. Our system takes as input a pair of stereo images and outputs segment proposals. For each region in the segmentation hierarchy, we extract geometric and CNN features to represent it. Those adjacent regions are iteratively merged, which is guided by a similarity network. We select those single and merged regions as object proposals and rank them based on a ranking network.

ing single, pair, triple and 4-tuple neighboring segments from the hierarchies. We then learn a ranking network to predict the objectness score of each region proposal. Our similarity and ranking network use a combination of learned deep features for appearance and designed geometric features for depth cue. While the similarity network predicts how likely two regions belong to the same object instance or the same background class, the ranking network estimates the overlap ratio with respect to the ground truth for each candidate region. The illustration of our system is shown in Figure 1.4.

3) Learning spatial transforms for refining object segment proposals. An alternative approach to generating better object proposals is to refine an initial set of object segments produced by existing methods. To realize this idea, we propose an efficient object segment refinement method that learns spatial transforms to improve the pixel-level accuracy of the object proposals. Our method takes both image and initial object masks as input, and predicts a spatial affine transformation in 2D image plane for each mask, which is then used to warp the corresponding mask into a more accurate object segment candidate.

To be more specific, we formulate the segment refinement as a regression problem, and build a deep network to predict the 2D affine transformation required for

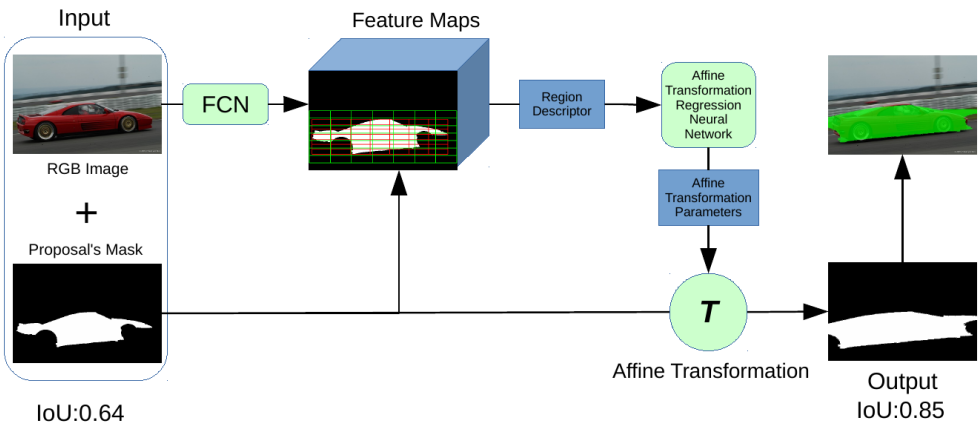


Figure 1.5: Model structure of our approach for learning spatial transforms for refining object segment proposals. Our system takes as input an image and initial segment proposals. It first extracts deep features to describe a segment and feeds the descriptor into a regression network to estimate an affine transformation. We then apply the affine transformation to the segment mask to obtain the warped mask.

improving the mask accuracy. Given the input image, we first extract a hypercolumn feature representation to represent the multi-scale image cues. On these feature maps, we design a novel mask pooling scheme that incorporates cues from both an initial object segment and its spatial context. The pooled features are fed into a four-layer neural network, which outputs affine transformation parameters for warping the object mask. To train the regression network, we precompute the affine transformations from the initial object masks to their corresponding groundtruth masks based on nonrigid registration, which are used as our regression targets. Our model structure is displayed in Figure 1.5.

4) Learning deep free-form deformation network for object-mask registration. As affine transformations can only capture coarse global deformations and the proposed previous entire system cannot be trained in an end-to-end fashion, there are limitations in the last approach. To overcome these limitations, we propose a deep free-form deformation (FFD) network to address the more general object-mask alignment problem and apply this network to the task of refining segment proposals. Given an input image containing the target object and an initial mask, our approach learns a non-rigid 2D transform that warps the mask onto the target object, which can generate much better aligned proposals than those obtained through affine transformations. To achieve this, we design a novel spatial transformer network that predicts a free-form deformation transform and applies the non-rigid transform to the input mask to generate a better alignment between the mask and object.

Specifically, we build a deep convolutional neural network consisting of two modules. The first module computes the convolutional feature maps from the input image, and extracts a feature representation of the image region covered by the mask. To encode the shape information of the initial mask and the image cues around

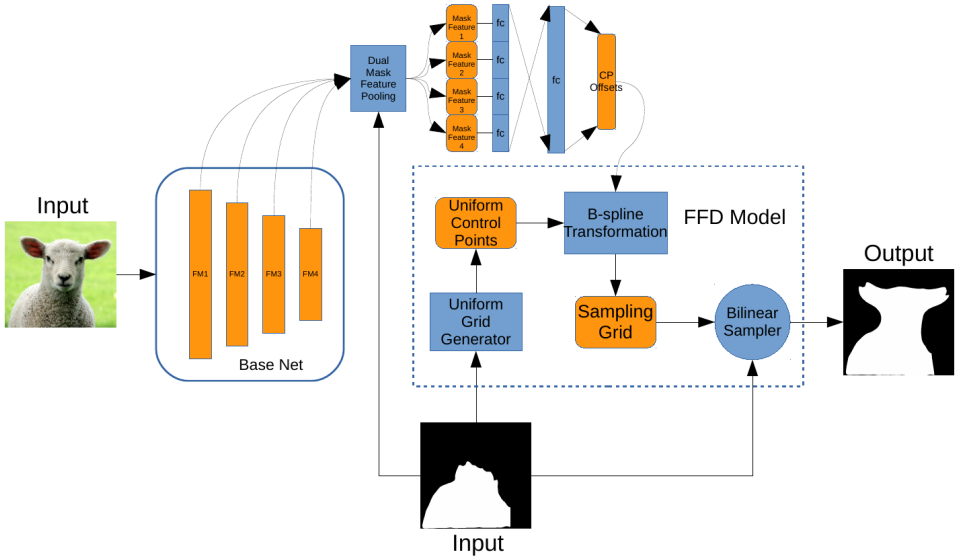


Figure 1.6: Overview of our deep FFD network for object-mask alignment. The entire network consists of two modules: the first computes the convolutional feature maps and extracts mask features using dual mask pooling, and the second predicts the FFD transform and warps the input mask onto the target object.

object, we develop a multi-level dual mask feature pooling method to capture the misalignment between the mask and object. Based on the multi-level features, the second network module predicts a free-form deformation (FFD) transform parameterized by the offsets of predefined control points through regression. It then applies the B-spline based FFD transform to the initial mask based on a grid generator and a bilinear sampler, which produces the final warped object mask. As those two network modules are differentiable, we can train the entire deformation network in an end-to-end fashion using a L_2 matching loss. An overview of our deep FFD network is shown in Figure 1.6.

1.4 Thesis Outline

The next chapter review literature relating to the research problems to be addressed in this thesis. We will first look at different vision tasks in which object proposals play an important part. Then we discuss early work on object proposal generation and categorize them according to the form of the proposal. Next, we briefly describe 3D reconstruction, mainly focusing on disparity estimation for stereo images. After that, we discuss the convolutional neural network (CNN) and feature representation learning. Following the discussion of the CNN, we review modern CNN-based approaches to object proposal generation. Finally, we discuss the work on modelling and learning spatial transforms, establishing the basis for warping the object proposals.

Chapter 3 addresses the problem of incorporating geometric information and semantic context into the object bounding box proposal generation for stereo images. Unlike existing methods which mostly rely on image-based or depth features to generate object candidates, we propose to incorporate additional geometric and high-level semantic context information into the proposal generation. Our method starts from an initial object proposal set, and encode objectness for each proposal using three types of features , including a CNN feature, a geometric feature computed from a dense depth map, and a semantic context feature from pixel-wise scene labelling. We then train an efficient random forest classifier to re-rank the initial proposals and a set of linear regressors to fine-tune the location of each proposal. Experiments on the KITTI dataset show that our approach significantly improves the quality of the initial proposals and achieves the state-of-the-art performance using only a fraction of original object candidates.

Chapter 4 proposes a learning-based object segment proposal generation method for stereo images. In contrast to existing methods which mostly rely on low-level appearance cue and hand-crafted similarity functions to group segments, our method makes use of learned deep features and designed geometric features to represent a region, as well as a learned similarity network to guide the grouping process. Given an initial segmentation hierarchy, we sequentially merge adjacent regions in each level based on their affinity measured by the similarity network. This merging process generates new segmentation hierarchies, which are then used to produce a pool of regional proposals by taking region singletons, pairs, triplets and 4-tuples from them. In addition, we learn a ranking network that predicts the objectness score of each regional proposal and diversify the ranking based on Maximum Marginal Relevance measures. Experiments on the Cityscapes dataset show that our approach performs significantly better than the baseline and the previous state-of-the-art.

Chapter 5 addresses the problem of object segment proposal refinement. In contrast to prior work that predicts binary segment masks from images, we take an alternative refinement approach to improve the quality of a given segment candidate pool. In particular, we propose an efficient deep network that learns 2D spatial transforms to warp an initial object mask towards nearby object region. We formulate this segment refinement task as a regression problem and design a novel feature pooling strategy in our deep network to predict an affine transformation for each object mask. We evaluate our method extensively on two challenging public benchmarks and apply our refinement network to three different initial segment proposal settings. Our results show sizable improvements in average recall across all the settings, achieving the state-of-the-art performances.

In Chapter 6, we address the general problem of object-mask registration, which aligns a shape mask to a target object instance. Prior work typically formulate the problem as an object segmentation task with mask prior, which is challenging to solve. In this work, we take a transformation based approach that predicts a 2D non-rigid spatial transform and warps the shape mask onto the target object. In particular, we propose a deep spatial transformer network that learns free-form deformations (FFDs) to non-rigidly warp the shape mask based on a multi-level dual mask feature

pooling strategy. The FFD transforms are based on B-splines and parameterized by the offsets of predefined control points, which are differentiable. Therefore, we are able to train the entire network in an end-to-end manner based on L_2 matching loss. We evaluate our FFD network on a challenging object-mask alignment task, which aims to refine a set of object segment proposals, and our approach achieves the state-of-the-art performance on the Cityscapes, the PASCAL VOC and the MSCOCO datasets.

Chapter 7 summarizes the main content of this thesis and discusses future directions for research in object proposal generation.

1.5 Thesis Contributions

Object proposal generation is a research area with rapid progresses. In this thesis, we extend the proposal generation from single modality to multi-modalities, introduce the representation learning and similarity learning into the segment proposal generation, and design deep networks to refine segment proposals through warping them. The thesis contributions are summarized as follows:

- We propose a new pipeline to generate object bounding box proposals for stereo images based on additional geometric and semantic context cues. We also introduce the location regression into the object bounding box proposal generation.

  This work has been published at ICIP 2016 [43]

- We develop an alternative deep learning approach to the object segment proposal generation for stereo images. The proposed framework allows us to train a similarity network to make use of both learned deep features and designed geometric features to group image pixels into meaningful objects. We also design a ranking network to evaluate the quality of each segment proposal.

  This work has been published at ACCV 2016 [44]

- We propose a novel refinement method that learns spatial transforms for improving the quality of object segment proposals. We design and train an efficient deep network to predict the instance-level affine transformations, based on hypercolumn feature and mask pooling, which can be used to warp the corresponding mask into a more accurate object segment candidate.

  This work has been published at WACV 2017 [45]

- We propose a deep free-form deformation network that aligns a shape mask to a target object instance. This novel spatial transformer network predicts a free-form deformation (FFD) transform, and applies the non-rigid transform to the input mask to generate a better alignment between the mask and object. This network can capture highly non-rigid deformations between a shape mask

and its corresponding object, and is fully differentiable that can be trained in an end-to-end manner.

This work has been published at ICCV 2017 [46]

Chapter 2

Literature Review

Object proposal generation aims to produce a set of high-quality object candidates that have high recall rates, low counts and good boundary coverage in an image. It has evolved rapidly since being recently proposed recent years and has become a core component in many vision tasks, like object detection and object instance segmentation.

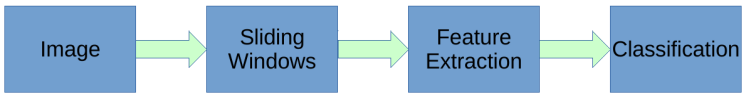
In this chapter, we will first review different vision tasks in which object proposals play an important part. Then we discuss early work on object proposal generation and categorize them according to the form of the proposal. One is the bounding box proposal generation, while the other is the segment proposal generation.Next, we briefly describe 3D reconstruction, mainly focusing on disparity estimation for stereo images. After that, we discuss the convolutional neural network (CNN) and feature representation learning. Following the discussion of the CNN, we review modern CNN-based approaches to object proposal generation. Finally, we discuss the work on modelling and learning spatial transforms, establishing the basis for warping the object proposals.

2.1 Object Proposal in Computer Vision

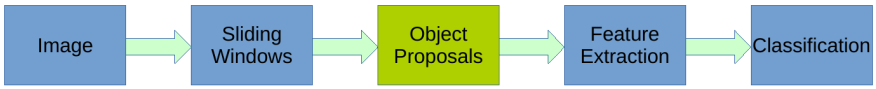
Object Detection Object proposal originates from the practical need of object detection. Object detection, one of the fundamental challenges in computer vision, aims to detect instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images or videos. The dominant framework for object detection over the past decade has been the sliding window paradigm (Figure 2.1(a)), in which object classification is performed at every location and scale in an image [2, 19, 20].

The works in [19, 20] represent the typical sliding-window object detection system. They extract HoG features for every window at different image scales and then classify each window using learned classifiers. Since the extraction of HoG features is not efficient and millions of windows need to be processed, the time consumed by such a pipeline in the testing stage tends to be quite high.

By contrast, Viola and Jones [2] propose a rapid cascade of classifiers for face detection (see Figure 2.2). They first use extremely simple and efficient classifiers to discard most of those windows that are unlikely to contain a face and then process



(a) The pipeline of early object detection under sliding window paradigm



(b) The pipeline of modern object detection using object proposals

Figure 2.1: Comparison between early stage and modern object detection

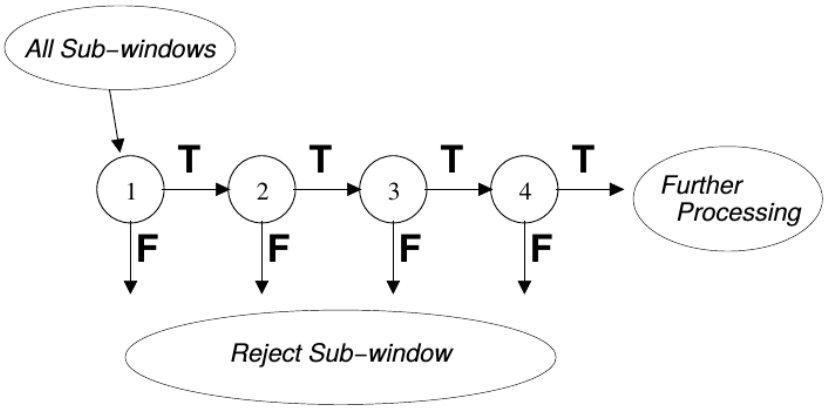


Figure 2.2: Schematic depiction of the detection cascade proposed in [2]

.

the left promising windows with a sequence of more complex classifiers. This strategy, which is quite similar to the core idea of object proposal, dramatically increases the speed of the face detection system.

With the development of object detection, more and more complex classifiers are used. This brings the improvement on detection accuracy, but inevitably increases the computational burden. To reduce the runtime complexity, one approach is to decrease the cost of classification, such as using efficient implementation of kernel method [47, 48]. Another approach is to reduce the number of windows to be evaluated by the complex classifier [2, 49, 50]. This approach first runs a very efficient classifier over all windows and only keeps a few highly scored windows for evaluation by the complex classifier. This idea leads to the emergence of the object proposal [4]. Because of the successful use of object proposals, the traditional pipeline of object detection has been less broadly used than its modern version, illustrated in Figure 2.1(b), where object proposal generation plays a core role.

Particularly, in the breakthrough work of RCNN [32], which we will detail later, Girshich *et al*. propose a CNN-based model for object detection that applies a CNN classifier on each of object proposals generated by an existing method. This approach

achieves a significant gain in object detection performance compared to classic sliding window methods. Since then, most state-of-the-art object detection algorithms rely on object proposals, which facilitates the fast advance of object proposal generation. The improved version of RCNN, faster RCNN [10] further proposes a regional proposal network (RPN) to generate object proposals, which are classified by another deep network adopted from fast RCNN [8]. The rapid progress in object detection consolidates the core role of object proposals and pushes object proposal generation to advance further.

Object Instance Segmentation Compared with object detection, object instance segmentation, which aims to segment out every instance of each object category in an image, is a more challenging task. It requires identifying each object instance in the form of a mask rather than a bounding box, providing more accurate localization information of an object than the object detection.

Prior work on instance segmentation usually starts from bounding box detections. Yang *et al*. [51] combine top-down deformable shape priors with bottom-up grouping constraints to produce high-quality object segmentations, based on the output of object detection. Parkhi *et al*. [52] use the template-based model to detect a distinctive part for the class and then detect the rest of that object via segmentation using image specific information learned from that part. In [53], Dai *et al*. first detect the object using a modified version of the DPM detector [20] and then predict which pixels are part of the object based on color and edge information. Fidler *et al*. [54] propose a DPM model that exploits region-based segmentation by allowing every detection to select a segment from a pool of segment candidates.

Instead of starting from the detector output, another method direction is to align a shape mask to object instances. Early work on level-set based segmentation starts from an initial contour and iteratively evolves the contour toward the target object by minimizing a functional energy function [55]. More recent approaches tend to use initial masks as a prior in inferring object segmentation. Kuettel *et al*. [56, 57] transfer segmentation masks from training windows that are visually alike to windows in the test image. He and Gould [58] develop an exemplar-based approach to the task of instance segmentation, in which a set of reference image/shape masks are used to find multiple objects based on discriminatively trained Exemplar-SVMs. Tighe and Lazebnik [59] employ the similar method for scene segmentation. [60] first generates a set of class-specific figure-ground segmentation masks for a human body, and then matches, aligns and fuses shape priors generated from data with these region masks to get a better segmentation of people. However, these methods share similar problems that fining a similar exemplar in the training set is always time-consuming and transferring a mask from training images to test images tends to generate inaccurate masks for the target objects. In Chapter 6, we propose a free-form deformation network that learns non-rigid transformations to register the shape mask onto the target object, which is more efficient and effective.

Current top-performing methods employ the similar pipeline to modern object detection approaches. Most of them use the CNN to score object segment proposals

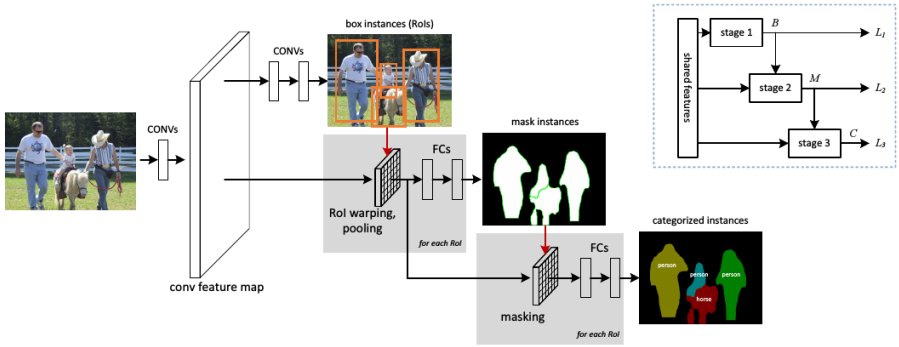


Figure 2.3: Multi-task Network Cascades for instance-aware semantic segmentation [3]. At the top right corner is a simplified illustration.

to do object instance segmentation. Hariharan *et al*. [33] use MCG [5] to generate segment proposals and then extract two types of CNN features from each segment, followed by segment classification and refinement. In [9], Dai *et al*. follow the similar pipeline to [33] but design a feature masking technique to extract CNN features for segment proposals. Dai *et al*. [3] further propose a multi-task network cascade for instance segmentation (see Figure 2.3 for their network architecture), in which the first two stages generate generic bounding box proposals as well as an object segment for each bounding box, and then classify these object segments in the final stage.

The proposal-based object instance segmentation paradigm enables this task to benefit from the rapid advance in object detection, as they share similar pipeline through the object proposals. This leads to the great progress in this area and also attracts a lot of attention into the research of object segment proposal generation.

2.2 Early Stage Object Proposal Generation

Although object proposal generation is a relatively new research topic in the context of deep learning, much progress has been made in this direction/area and a lot of algorithms have been proposed in recent years. In the early stage, most proposal approaches use hand-crafted low-level features or manually designed strategies to produce object proposals. Along with the widespread use of CNN in computer vision, the object proposal generation has also stepped into the CNN era, beginning to rely on learned feature representation. In this section we mainly review the early stage object proposal generation methods and divide them into two categories: object bounding box proposal generation and object segment proposal generation.

2.2.1 Object Bounding Box Proposal Generation

Object bounding box proposal generation usually takes a window-scoring pipeline. This paradigm assigns an objectness score to each candidate window according to

(a) (b) (c)

Figure 2.4: Idea property of an objectness measure. The objectness measure should score the blue windows, partially covering the objects, lower than the groundtruth windows (green), and even lower the red windows containing only stuff [4].

how likely it is to contain an object. The objectness score is in general output by a learned classifier or a designed score function. Given a score threshold, those windows that have scores higher than the threshold are kept as object proposals.

Objectness [4] is one of the earliest well-known proposal approaches. In this work, Alexe *et al*. propose a classification framework to compute an objectness score for each sliding window (see Figure 2.4). They compute the objectness score using multiple image cues, such as color, edge, location, size and the superpixels straddling. Among these cues, the superpixels straddling is most discriminative and has also been adopted in other approaches [63, 64]. The superpixels straddling cue measures for all superpixels s the degree by which they straddle a window w

$$SS(\omega, \theta_{SS}) = 1 - \sum_{s \in S(\theta_{SS})} \frac{min(|s \backslash \omega|, |s \cap \omega|)}{|\omega|} \tag{2.1}$$

where $S(\theta_{SS})$ is a set of superpixels obtained with a segmentation parameter θ_{SS}. For each superpixel s, Equation 2.1 calculates its area $|s \cap \omega|$ inside ω and its area $|s \backslash \omega|$ outside ω. The minimum of the two contributes to the sum in this equation.

Rahtu [61] builds on the idea of 'Objectness' and learns a cascade layer to efficiently rank proposals. Their algorithm starts with a large set of bounding boxes composed by superpixel windows and a large number of sampled boxes. Then they extract proposed objectness features from these initial windows. Finally, they estimate the objectness score for each candidate window using a structured output ranking objective function.

Zhang [62] also employs a cascade of classifier to generate proposals. They first train a set of ranking SVMs separately for each scale and each aspect-ratio of the window and then rank all proposals output from the first stage.

Bing [36] learns a very fast linear classifier based on image gradients and applies the classifier in a sliding window fashion. The prominent advantage of Bing is its high speed (1 ms/image on CPU) but its performance suffers in terms of object recall rate under high IoU thresholds.

Randomized SEEDS [63] exploits multiple superpixel maps to compute the objectness score for each of temporal windows. They use the intersection of several superpixel partitions to estimate the boundaries, based on which they design a simple object proposal metric similar to the superpixels straddling from Objectness [4]. This method, however, has a low computation efficiency.

EdgeBoxes [37] proposes a simple objectness score that measures the number of edges which are wholly contained in a bounding box. They compute the objectness score also in a sliding window pattern. They further design a search strategy to tune the density of bounding boxes according to the desired intersection over union (IoU) threshold. This method is elegant and has been widely used.

Chen [64] proposes a bounding box refinement method using the cue of superpixels straddling [4]. They first align initial bounding boxes with boundaries preserved by superpixels and then expand each bounding box according to the straddling degrees of superpixels. They achieve sizeable improvements over several existing methods.

3DOP [65] generates 3D object proposals for stereo images. They design an MRF framework to exploit object size priors, ground plane as well as several depth features to produce object proposals in 3D space. However, they focus on the class-dependent object proposals and do not consider the semantic context information.

Except 3DOP [65], early stage object bounding box proposal methods mainly work on the single modality, and focus on low- or mid-level image features, which are insufficient and ambiguous to predict the objectness score. On the other hand, objects existing the physical world need to satisfy certain geometric/physical constraints. For example, cars and persons have certain sizes and usually stand on supporting planes. Hence, geometric information tends to be a strong indicator of how likely a certain area contains an object. In addition, objects have close relations to their neighbouring object/background classes, such as cars usually occur on the road and bicycles tend to be on the ground plane. Semantic context can help to remove ambiguities caused by local appearance information. Therefore, we believe that the geometric information and the semantic context cues can benefit the proposal generation and further improve proposals' quality, which inspires our first work that generates object candidates with geometric feature and context cues for stereo images.

2.2.2 Object Segment Proposal Generation

Compared to bounding box proposals, object segment proposals are more informative, for providing better object localization and boundary alignment. Also, segment proposals can be easily transformed into bounding box proposals, simply by taking the tight bounding box enclosing the segment. However, generating segment proposals is more challenging than producing bounding box proposals. Early segment proposal methods generally take a perceptual grouping paradigm, where homogeneous perceptual pixels are grouped to form a segment candidate. The pixel grouping can

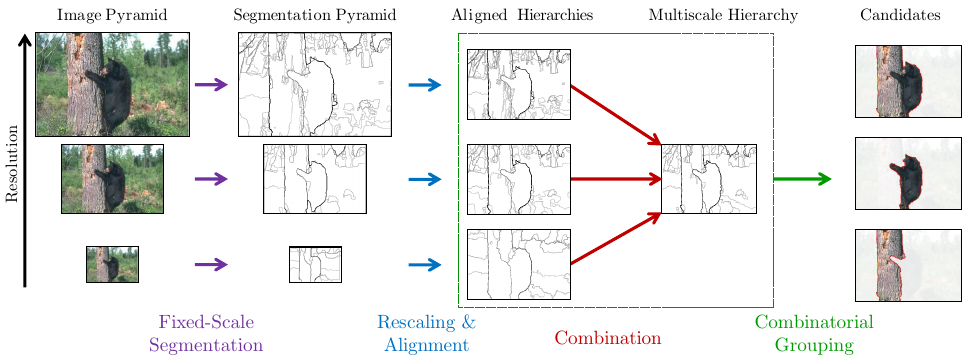


Figure 2.5: The pipeline of multiscale combinatorial grouping (MCG) [5].

be achieved by solving multiple segmentations with diverse seeds or by merging superpixels in multiple over-segmentations.

CPMC [31] places multiple seeds on an image grid and solves a constrained parametric min-cut problem for each seed to compute figure-ground object hypotheses. To reduce the redundancy of the proposals, they remove a large part of proposals through maximum relevance ranking and diversification. This approach outperforms significantly previous low-level segmentation methods, but solving a sequence of CPMC problems is quite inefficient.

Endres [66] generates a segmentation hierarchy from occlusion boundaries and iteratively performs graph-cut on superpixel affinity graphs with a variety of parameters to produce segments. These initial segments are then ranked based on multiple cues. This method achieves good boundary alignment but it is time-consuming.

SelectiveSearch [40] iteratively groups superpixels into a segmentation hierarchy and selects the segments from the hierarchy as object proposals. In order to diversify the pool of segments, they generate multiple segmentation hierarchies using a variety of grouping criteria and complementary colour spaces. Because of its good performance in terms of recall and quality, it has been broadly used in many top-performing object detection algorithms as the proposal method. But this method lacks an effective way of estimating proposal importance.

RandomizedPrim's [67] can be seen as a modified version of SelectiveSearch. It further introduces a randomised superpixel merging process which learns superpixel similarity measures to diversify the grouping results. However, it cannot attain high object recalls.

GOP [42] begins with a segmentation hierarchy and places multiple seeds for a geodesic distance transform on the image by learned classifiers. For each seed they generate foreground and background masks, for which they then compute a signed geodesic distance transform over the image. Finally, they extract a set of object pro-

posals by identifying certain level sets of the distance transforms. This method, however, is unable to assign scores to proposals.

Rantalankila [68] combines the superpixel merging with graph cut segmentations. They first generate a part set of proposals by taking regions from a segmentation hierarchy and then perform CPMC [31] on an intermediate level of the hierarchy to obtain a complementary set of proposals. The results, however, are inferior to state-of-the-art.

Rigor [69] is similar to CPMC [31] but speeds it by pre-computing a graph for solving multiple grape cut problems. But the higher efficiency is obtained at the cost of lower object recalls.

LGO [70] shares the same spirit of SelectiveSearch [40], except that they use different features and train a Random Forest to guide the grouping process. This brings better object recalls, indicating learning a grouping process is beneficial to proposal generation.

MCG [5] generates multi-scale segmentation hierarchies based on [71] and takes a subset of singletons, pairs, triplets and 4-tuples from the hierarchies as object proposals. MCG shows high performance in terms of recall and quality and is widely used as the proposal method in many instance segmentation systems. However, this algorithm runs slowly. The overview is shown in Figure 2.5.

Lee [72] proposes a parametric energy function for structured multiple output learning and combines multiple mid-level cues to yield segment proposals by grouping superpixels under this framework. But the performance of this method is inferior to the state-of-the-art in terms of object recall rate and object proposal accuracy.

Wang [73] follows the similar paradigm to SelectiveSearch, but learns complementary superpixel merging strategies so that errors generated in one merging strategy can be corrected by the others. Results show that additional learned grouping process enables this method to achieve better object recalls.

LPO [41] trains an ensemble of complementary figure-ground segmentation models and performs bottom-up segmentation by applying each model on the image to produce segment proposals. This algorithm can recall more small objects at the cost of complex learning process.

Bleyer [74] designs an iterative labeling strategy to segment object proposals from stereo images. It makes use of depth information to identify the extent of objects. It can work with stereo images, however, it is computationally expensive.

Similar to bounding box proposal methods, except the work by Bleyer *et al*. [74], early segment proposal approaches also rarely exploit the geometric cue and semantic information to further improve the quality of the proposals. Besides, most superpixel grouping methods mainly rely on low-level image features to compute the affinity between superpixels, which may lead to inaccuracy. Using color, size

and edge *etc*. low-level features to represent a superpixel tends to bring ambiguity into the perceptual grouping. For example, an image of a cat with a colourful coat may be segmented into many meaningless parts just according to color information and these parts are hardly to regroup into a cat just using low-level image features. On the other hand, high-level semantic information provides more stable cue to merge superpixels into a meaningful object. Although the parts of the coat of the cat have different colours, they all belong to this cat. Therefore, incorporating high-level semantic features into the descriptor of a superpixel will benefit the grouping process. In addition, the early superpixel grouping methods seldom directly learn the similarity measures of superpixels from the data. Instead, most of them use pre-defined similarity metrics, such as boundary strengths, to guide the grouping process, which lacks global context support and may be ineffective. We believe that a learning method for computing the similarity will be more effective, as it can better adapt to the data distribution. From these arguments, we propose a deep learning-based segment proposal approach that learns to group neighboring image regions into meaningful objects for stereo images in our second work.

2.3 3D Reconstruction

To extract geometric cues for an object in stereo images, we need first reconstruct the scene from stereo images. As this is not our focus, we briefly introduce the methods used in our work to estimate the disparity map and simply describe the calculation of converting the disparity map into the point cloud.

2.3.1 Disparity Estimation

Given a pair of rectified stereo images, estimating the disparity map generally needs the following four steps [75]:

1 matching cost computation;

2 cost aggregation;

3 disparity computation / optimization; and

4 disparity refinement.

In our work, we mainly employ the Semi-Global Matching (SGM) [76, 77] method to compute the disparity map. The SGM method is based on the idea of pixel-wise matching of mutual information and approximating a global, 2D smoothness constraint by combining many 1D constraints. Specifically, according to the 4-step pipeline, SGM first computes the matching cost based on mutual information [78] for each pixel. To ensure the smoothness of estimated disparities, an additional constraint that penalizes discontinuities is added to the matching cost. In the second step of cost aggregation, SGM aggregates matching cost from all directions equally by summing the costs of all 1D minimum cost paths that ends in that pixel. After

that, the disparity map is determined by selecting for each pixel the disparity that corresponds to the minimum cost. Finally, to obtain sub-pixel estimation, a quadratic curve is fitted via the neighboring costs.

To preserve the boundary of different image segments and smooth the disparity map got from SGM method, [77] further inputs the results got from SGM into a slanted plane smoothing system [79]. We use the output from this system as our final disparity map.

Even though post-processing stages that impose smoothness and handle disparity discontinuities and occlusions have been proposed in most algorithms for disparity estimation, the produced disparity maps still tend to be quite noisy due to occlusions or textureless regions. In order to extract robust geometric features from those noisy depth maps, we have manually designed different groups of 3D features to encode object geometric properties, which we will detail in specific work.

2.3.2 Point Cloud Generation

Given the disparity map and calibration parameters of stereo cameras, we can easily covert the disparity map into a dense depth map and then a point cloud representation of the scene.

Formally, let f be the focal length of the cameras, let b be the distance between the stereo cameras (the stereo baseline) and let d be the disparity. Then the depth of the point in the scene under the coordinate system defined by the reference image, z, can be computed by the following equation

$$z = \frac{b}{d} f \tag{2.2}$$

With the depth z calculated, the width x and height y of this point from the center of the scene can be got by below equations

$$x = \frac{u - u_c}{f} z, \quad y = \frac{v - v_c}{f} z \tag{2.3}$$

where u and v are the pixel row and column locations in the 2D image respectively, while u_c and v_c denote the center of the 2D image.

2.4 Convolutional Neural Networks

Recently, the Convolutional Neural Network (CNN) has swept the computer vision community, with state-of-the-art performance for many vision tasks [6, 32, 80, 81]. The object proposal generation can also benefit from the CNN, and in this section we review those studies that are related to feature representation learning and deep feature extraction. In addition, our methods are built on top of several deep learning techniques, which will be briefly described simultaneously.

In 2012, Krizhevsky *et al*. [6] showed that they achieved a significant improvement

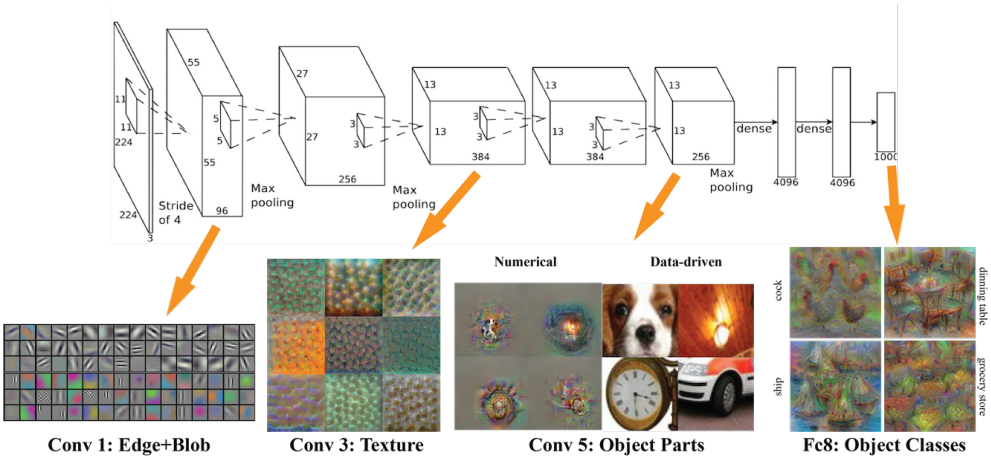


Figure 2.6: Illustration of the typical structure of a CNN (AlexNet [6]).

on the ImageNet classification benchmark [82, 83] by training a large convolutional neural network (CNN) on 1.2 million labelled images. This rekindled the interest of the community in the CNN. Since then, the CNN has developed quite rapidly and has been the main tool in many vision tasks like object detection [32, 8, 10], semantic segmentation [80, 84] and optical flow estimation [81], leading to substantial improvements on these tasks.

The great success of the CNN comes from that it can learn highly discriminative, yet invariant feature representations from big data. A typical modern CNN (see Figure 2.6) [6] usually consists of multiple convolutional layers, max-pooling layers, ReLU non-linear layers and fully connected (fc) layers which are usually followed by drop-out regularization layers. Its multiple layers enable it to represent an image by from a combination of low-level features to highly abstract semantic information. The feature maps output from higher layers represent the semantic information of the image, as the activations on higher layers have a broader receptive field and can encode a distributed representation of color, edge, shape and texture of the image which are captured by lower layers. Particularly, the very deep networks like VGG [85] and GoogLeNet [86] boost the representation ability even further, performing considerably better than AlexNet [6] on ImageNet. At the same time, these models can learn sufficiently general image features, which can be transferred to many different tasks, promoting them greatly through supervised transfer learning. This has greatly benefited many vision tasks [32, 8, 80, 10], as well as the object proposal generation [38, 11, 3, 12].

The success of the CNN also comes from a variety of proposed techniques for making effective use of CNN features. In the following, we briefly introduce those techniques that are related to our work.

- **CNN as Fixed Feature Extractor.** This method takes models pre-trained on ImageNet for image classification as generic feature extractor. It usually uses the output from fully connected (fc) layers as features, though those feature

maps output from convolutional layers can also be used to extract features. Jia *et al*. [87] show that features taken from fc layers of a pre-trained model have sufficient representational power and generalization ability. By using the features, they outperform previous state-of-the-art approaches with traditional hand-crafted features on several recognition tasks including scene classification, fine-grained subcategorization and domain adaptation.

- **Fine-tuning.** This strategy is to fine-tune all or part of the weights of the pre-trained network on the new dataset. This approach is the mainstream at present to leverage those pre-trained models, like AlexNet and VGG trained on ImageNet. To train a deep CNN from scratch is expensive because it needs a huge amount of labelled data and a lot of time to learn the model weights. But the CNN models pre-trained on the ImageNet are limited to the task of image classification. So most systems relying on CNN models start with pre-trained networks and then transfer the networks to the target task and dataset using supervised fine-tuning. Many popular vision systems [32, 8, 10, 3, 11, 12] have shown the effectiveness of fine-tuning.

- **Fully Convolutional Network.** The fully convolutional network (FCN) [80] is proposed for extracting features for pixels in order to adapt the CNN in semantic segmentation. The basic idea behind a fully convolutional network is that it is 'fully convolutional', that is, all of its layers are convolutional layers. FCNs do not have any of the fully connected layers at the end which are actually transformed into convolutional layers. This helps generate feature maps with the same height and width as the input image in the final output layer. Then each unit in the feature maps can represent a pixel in the input image, making it more convenient to extract features for a pixel, a bounding box or a segment.

- **Hyerpercolumn.** Hariharan *et al*. [7] define the Hypercolumn feature (see Figure 2.7) at a pixel as the vector of activations of all CNN units above the pixel. They extract Hypercolumn features for different vision tasks, achieving sizeable improvements over baselines. The improvements obtained by Hypercolumn features stem from the complementarity of CNN features output from different layers. The feature maps generated from the higher layers mainly capture semantics but they are usually too coarse spatially, while the feature maps produced in earlier layers mainly encode the low- and mid-level information such as the edge, color, shape and location, which are complementary to the higher feature maps.

- **RoI Pooling.** As a variant of the spatial pyramid pooling layer used in SPP-nets [88], region of interest (RoI) pooling [8] layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of $H \times W$. RoI max pooling works by dividing the $h \times w$ RoI window into an $H \times W$ grid of cells of approximate size $h/H \times w/W$ and then max-pooling the values in each cell into the corresponding output grid cell. In fast RCNN [8] (see Figure 2.8), a modified version of RCNN, RoI pooling helps

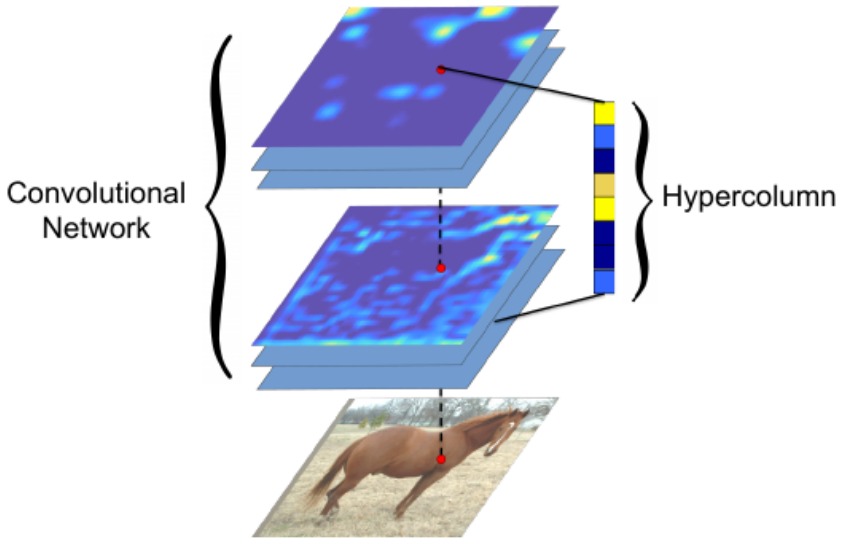


Figure 2.7: Depiction of hypercolumn representation. The hypercolumn feature at a pixel is the vector of activations of all units that lie above that pixel [7].

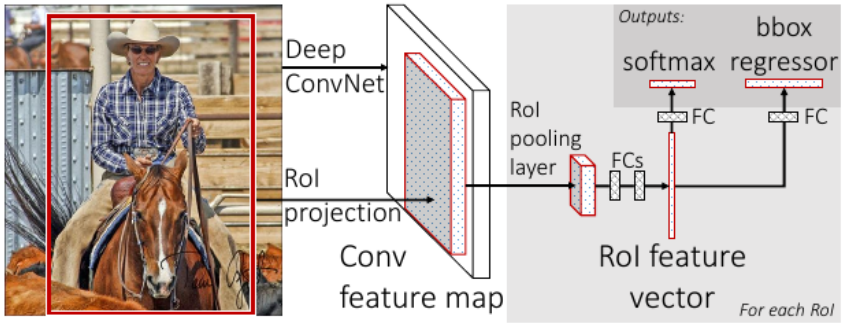


Figure 2.8: Fast RCNN architecture [8].

the detection network to train in an end-to-end fashion, improving significantly not only the training speed but also the detection performance.

- **Feature Masking.** In order to adapt the classification networks to the task of object instance segmentation, Dai *et al*. [9] develop a convolutional feature masking technique to form features for a segment proposal. Given the CNN feature maps and a segment proposal, they project the segment mask to the domain of the last convolutional feature maps and extract the segment features by multiplying the feature maps and the projected mask. An illustration can be found in Figure 2.9. This unifies the pipelines of object detection and object instance segmentation using the CNN, and benefits the latter task much.

Our algorithms are built on these techniques. We use the pre-trained FCN models

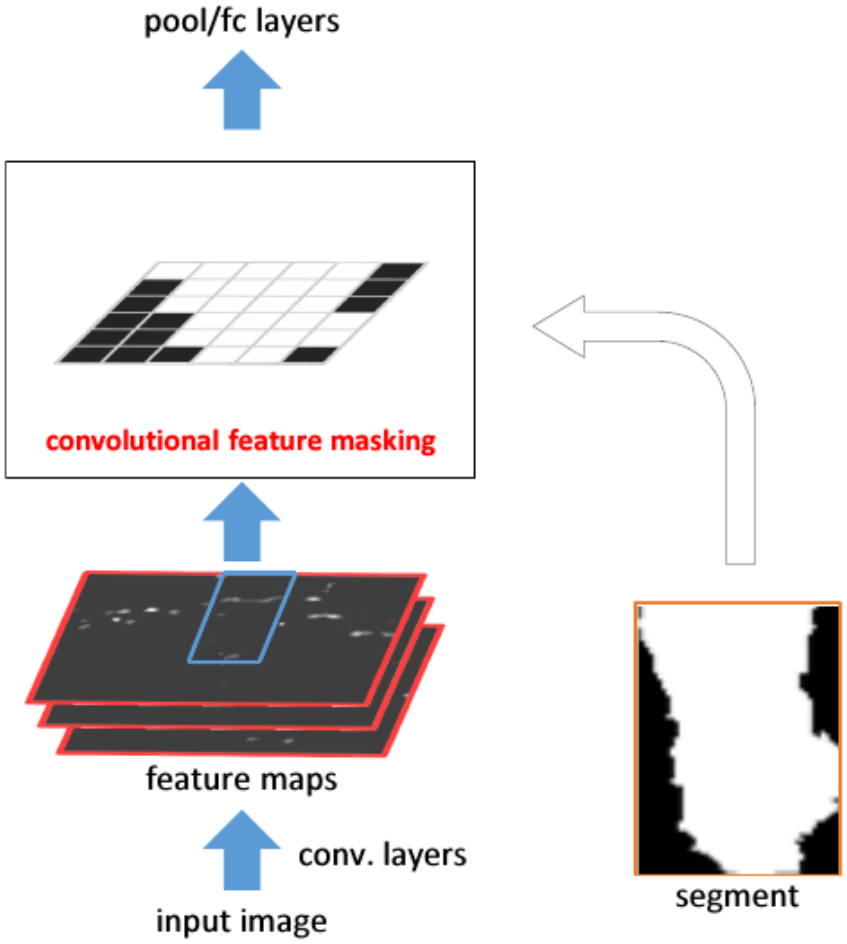


Figure 2.9: Illustration of the convolutional feature masking [9].

as our backbone networks and fine-tune their parameters for our tasks. We rely on RoI pooling, feature masking and Hypercolumn features or their variants proposed by us to extract our task-specific CNN features.

2.5 CNN-based Object Proposal Generation

Along with the popularity of the CNN in computer vision, object proposal generation also steps into the deep learning era. At present, most state-of-the-art proposal methods build on top of the CNN models. In this section, we mainly review recent advances in object proposal generation with CNNs.

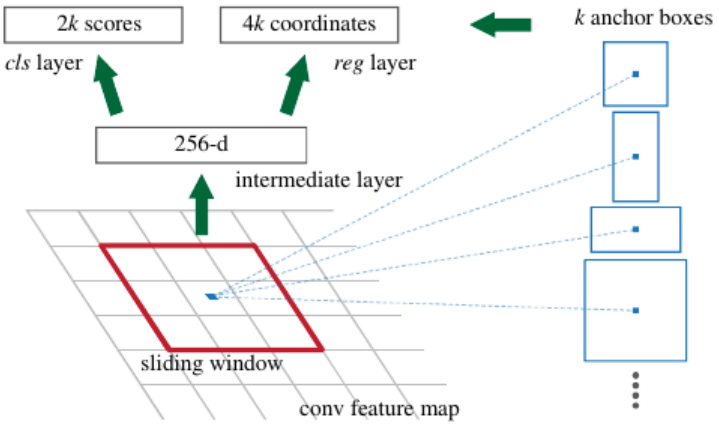


Figure 2.10: Illustration of the RPN proposed in faster RCNN [10].

2.5.1 Object Bounding Box Proposal Generation

DeepMultiBox [89] formulates the proposal generation as a regression problem. They train a CNN to simultaneously predict the coordinates and the objectness scores of bounding boxes. Their network architecture is similar to the AlexNet [6] and can only process one single image crop at test time.

DeepBox [38] learns a CNN to re-rank an initial set of bounding box proposals generated by EdgeBoxes [37]. They argue that the semantics learned by the network can benefit the prediction of the objectness score. They employ the network architecture proposed in fast RCNN [8], which helps them improve the processing speed significantly.

DeepProposal [39] refines a pool of selected windows in a coarse-to-fine manner. They start from selecting a set of dense proposals from the last convolutional layer, gradually remove irrelevant boxes using features coming from earlier layers and finally refine the localization of the left proposals through the refinement mechanism used in EdgeBoxes [37].

RPN [10] The work of faster RCNN proposes a region proposal network (RPN) to fast generate proposals, shown in Figure 2.10. In the RPN, a few pre-defined reference boxes are regressed into different locations with new aspect ratios. This is done by two sibling fully connected (fc) layers that take the features extracted on the last convolutional layer as input. One branch of the fc layers outputs the objectness scores for proposals, while the other predicts the locations.

Again, these CNN-based methods mainly work in the RGB image space and are not suitable for stereo images. It is still interesting to explore how the geometric information and semantic context can benefit the object proposal generation.

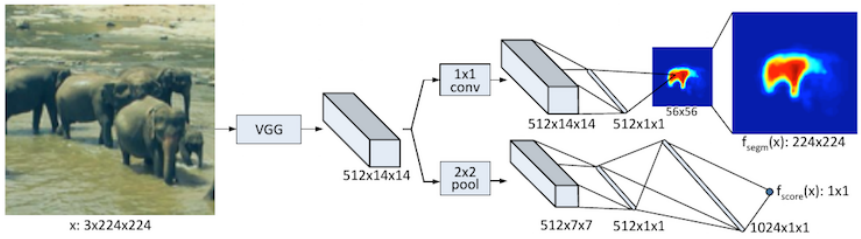


Figure 2.11: DeepMask network architecture [11].

2.5.2 Object Segment Proposal Generation

DeepMask [11] learns a CNN that predicts the objectness score for an image patch and simultaneously maps the image patch into a binary object segmentation mask. Its architecture is shown in Figure 2.11. This model is run convolutionally on the image to produce object segment proposals. Their experimental results show that DeepMask beats previous object segment proposal methods by a large margin. But it cannot generate high-quality masks that accurately align with object boundaries.

MNC [3] (see Figure 2.3) shares the similar pipeline to DeepMask for generating segment proposals. The difference is that MNC regresses object masks from bounding box proposals generated by the RPN [10], instead of applying the mapping to dense sliding windows. This improves the efficiency but may lead to truncated object boundaries.

SharpMask [12] proposes an object mask refinement network based on DeepMask. It starts from a coarse 'mask encoding' generated by DeepMask and refines this mask encoding in a top-down pass using features at successively earlier layers to produce a sharper object mask with better boundary alignment. An overview of its architecture is shown in Figure 2.12.

These CNN-based object segment proposal methods mainly employ an image-to-mask mapping solution. Even though they obtain substantial improvements over the previous approaches, however, learning such a direct image-to-mask mapping has shown to be challenging, which usually produces object masks lacking good boundary alignment and requires post-processing to improve their quality. Except SharpMask, few attempts have been made to improve the quality of initial segment proposals. Hence, we propose to refine the initial segment proposals through explicitly modelling the transformation between proposal mask ant its ground truth.

2.6 Modelling and Learning Spatial Transforms

To improve the quality of object proposals, an alternative approach is to learn spatial transforms to warp them by changing their locations or shapes, moving them closer

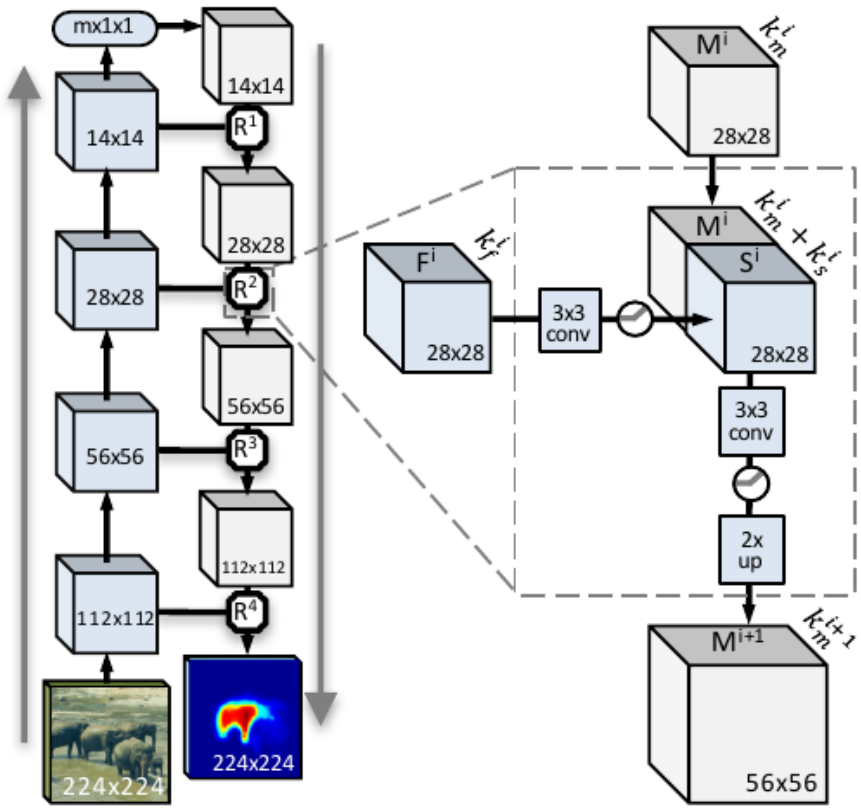


Figure 2.12: SharpMask network architecture [12].

to their ground truth. In this section, we investigate those works regarding modelling and learning spatial transforms.

2.6.1 Bounding Box Regression

In object detection, bounding box regression is an important step to improve localization accuracy. The classic object detection method DPM [20] learns a set of linear least-square regressors to map a feature vector to the coordinates of the bounding box. Inspired by this work, [32, 8, 10] also learn to predict a new bounding box for the detection using class-specific bounding box regressors, shown in Figure 2.13. Formally, denotes the bouning box by its center coordinates, its width and height, $\{B_x, B_y, B_w, B_h\}$. Similarly, the groundtruth box is denoted as $\{G_x, G_y, G_w, G_h\}$. The regression targets $\{T_x, T_y, T_w, T_h\}$ for the training pair (B, G) are defined as follows,

$$\begin{aligned} T_x &= (G_x - B_x)/B_w, \quad T_y = (G_y - B_y)/B_h \\ T_w &= \log(G_w/B_w), \quad T_h = \log(G_h/B_h) \end{aligned} \tag{2.4}$$

Given the regression targets and features extracted for the bounding box, location regressors like linear regressors with weights β can be learnt by by minimizing the

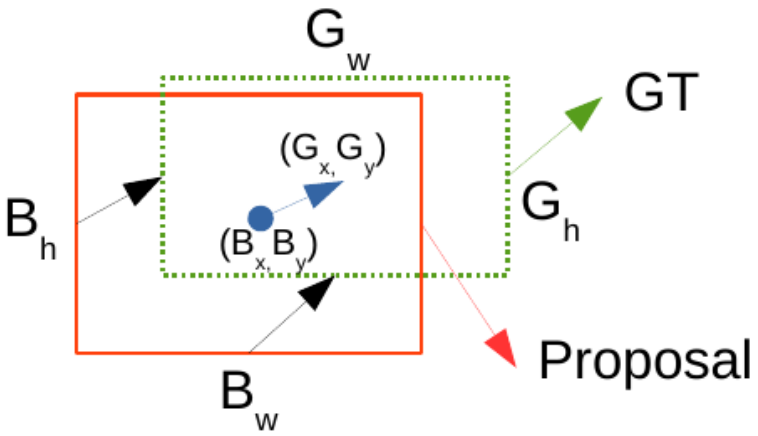


Figure 2.13: Illustration of bounding box regression. The learned regressors predict the offsets between the detection box (red) and the groundtruth box (green).

.

regularized least squares objective:

$$\beta = \arg\min_{\widehat{\beta}} \sum_{i=1}^{N} (T^i - \widehat{\beta}^T f(B^i))^2 + \lambda \parallel \widehat{\beta} \parallel^2, \tag{2.5}$$

where $f(B^i)$ denotes the feature extracted for the bounding box B^i, and λ is the weight for the regularization term.

Bounding box regression has been widely used in object detection, but it is class-specific and is limited to simple spatial transforms. Hence, we propose to exploit class-agnostic regressors in object proposal generation to refine bounding box locations. At the same time, inspired by bounding box regression, we come up with the idea of learning complex spatial transformations to warp segment masks into better object proposals.

2.6.2 Spatial Transformer Network

Recently, learning spatial transforms based on deep networks has been explored in a variety of problem settings. The spatial transformer network (STN) [13] learns an affine transformation to spatially warp feature maps into the canonical view to improve the classification accuracy. The STN consists of three parts, illustrated in Figure 2.14. A localization network takes the input feature map U (of size (H, W)) and outputs the parameters θ of the spatial transformation T_θ. A grid generator uses the parameters to create a sampling grid G. Finally, a sampler takes the feature map U and the sampling grid G as inputs to produce the warped feature map V. These three modules are all differentiable, allowing the STN to be trained in a end-to-end fashion.

Here, we detail the process of loss gradients flowing back to the input feature map, the sampling grid coordinates and the transformation parameters in STN, es-

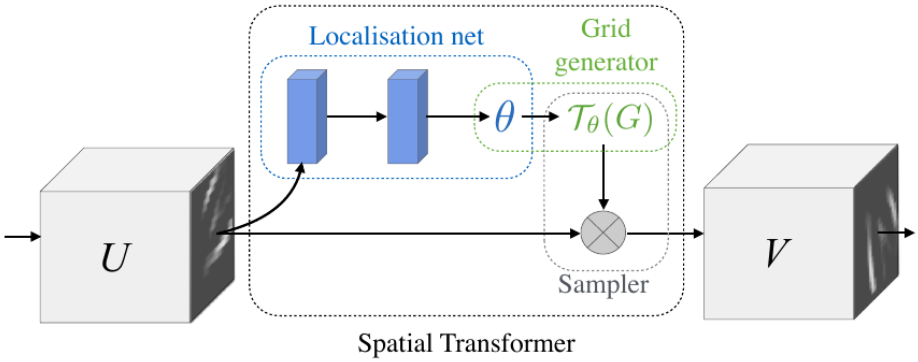


Figure 2.14: The architecture of a spatial transformer module [13]. U represents the inpute feature map, while V is the warped output feature Map.

.

tablishing the basis of gradients back propagation for our proposed free-form deformation network. Formally, denote the normalized coordinates in the input feature map U as (x_i^s, y_i^s) and the normalized coordinates in the output feature V as (x_i^t, y_i^t). At the same time, let U_{nm}^c be the value at location (n, m) in channel c of the input, and V_i^c be the output value for pixel i at location (x_i^t, y_i^t) in channel c. Suppose we employ bilinear interpolation to generate the sampling grid, then

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c max(0, 1 - |x_i^s - m|) max(0, 1 - |y_i^s - n|) \tag{2.6}$$

To allow back propagation of the loss through this sampling mechanism, we can define the gradients with respect to U and G. For bilinear sampling (2.6), the partial derivatives are

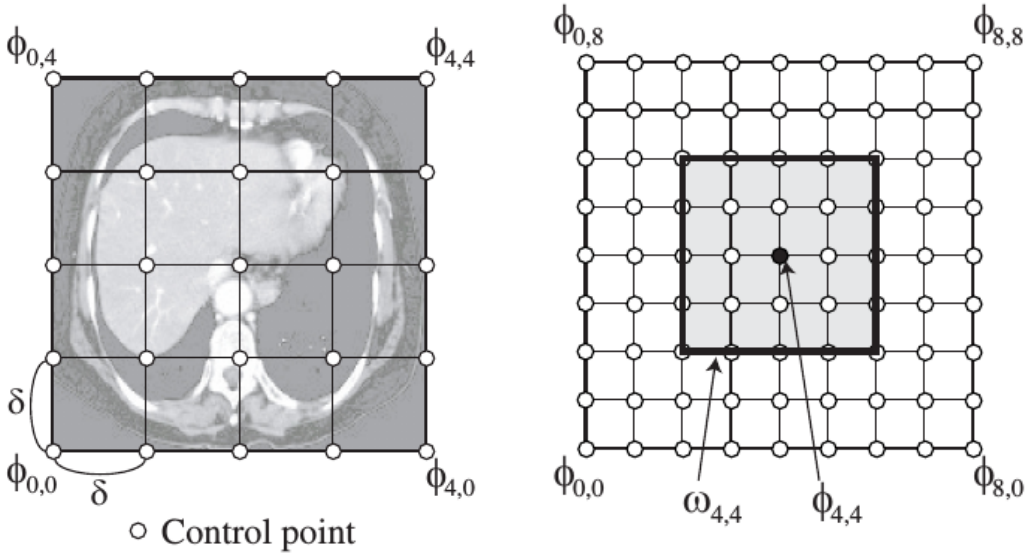
$$\frac{\partial V_i^c}{\partial U_{nm}^c} = \sum_n^H \sum_m^W max(0, 1 - |x_i^s - m|) max(0, 1 - |y_i^s - n|) \tag{2.7}$$

$$\frac{\partial V_i^c}{\partial x_i^s} = \sum_n^H \sum_m^W U_{nm}^c max(0, 1 - |y_i^s - n|) \begin{cases} 0 & if \quad |m - x_i^s| \geq 1 \\ 1 & if \quad m \geq x_i^s \\ -1 & if \quad m < x_i^s \end{cases} \tag{2.8}$$

and similarly to (2.8) for $\frac{\partial V_i^c}{\partial y_i^s}$. Given the transformation T_θ, we know that

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_\theta(G) = T_\theta \begin{pmatrix} x_i^t \\ y_i^t \end{pmatrix} \tag{2.9}$$

Then $\frac{\partial x_i^s}{\partial \theta}$ and $\frac{\partial y_i^s}{\partial \theta}$ can be derived from this equation (2.9). Thus, the loss gradients can back propagate from the output feature map to the input feature map through the differentiable sampling mechanism, making it possible to train the spatial trans-



(a) A mesh of control points (b) An active control point

Figure 2.15: B-spline free-form deformations (FFDs). (a) Deformations of a floating image are performed by manipulating an overlaying mesh of control points and (b) a control point affects points only inside its $4\delta \times 4\delta$ neighborhood domain. Images taken from [14]

former network in an end-to-end fashion.

Built on the mechanism of the STN, [90] proposes a deep deformation network for efficient object landmark localization, while [91] introduces a WarpNet to match images of objects, from which it builds single-view reconstruction. Different from STN, both of them employ the thin-plate spline (TPS) [92] transformations as the transform model.

2.6.3 Free-Form Deformation Model

Apart from the affine transformation and the TPS transformation, a more powerful tool to model spatial transforms is the free-form deformation (FFD) model based on B-splines [93], which has been widely used in medical image registration [94] and shape registration [95]. The basic idea of the FFD is to deform an object by manipulating an underlying mesh of control points (see Figure 2.15(a)). The control points act as parameters of the FFD model and determine the deformation being modelled. Formally, let Φ be a 2-D mesh of control points and $T : (x, y) \mapsto (x', y')$ be a pointwise transformation of any location (x, y) in target image F to the location (x', y') in the source image R. Given a mesh of control points $\phi_{i,j}$ with uniform

spacing δ pixels, the non-rigid transformation T by B-spline functions is defined by

$$T_{(x,y)} = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(u) B_m(v) \phi_{i+l,j+m} \tag{2.10}$$

where $i = \lfloor x/\delta \rfloor - 1$, $j = \lfloor y/\delta \rfloor - 1$, $u = x/\delta - \lfloor x/\delta \rfloor$, $v = y/\delta - \lfloor y/\delta \rfloor$, and B_l represents the l-th basis function of cubic B-splines [93]:

$$\begin{aligned} B_0(u) &= (1-u)^3/6, \quad B_1(u) = (3u^3 - 6u^2 + 4)/6 \\ B_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, \quad B_3(u) = u^3/6 \end{aligned} \tag{2.11}$$

From Equation (2.10), we note that the B-spline based FFD are locally controlled, in contrast to the affine transformation and the TPS transformation, as each control point $\phi_{i,j}$ affects only its $4\delta \times 4\delta$ neighborhood (illustrated in Figure 2.15(b)), making them computationally efficient. This also shows that the FFD can describe highly local and non-rigid transformations, which is required for capturing the complex non-rigid deformations between object mask and its ground truth. Additionally, the degree of non-rigid deformations can be controlled by changing the resolution of the mesh of control points Φ. A larger spacing of control points allows modelling of global and coarse deformation, while a small spacing of control points allows modelling of local and fine-grained deformation.

One recent work that takes FFD as the deformation model in learning spatial transformations is [96], in which they design a volumetric CNN that predicts deformation flows to get specified object shapes in 3D. They show impressive results in the 3D shape deformation task. By contrast, we mainly focus on the 2D class-agnostic object mask deformation.

2.7 Datasets and Evaluation

There are several datasets that provide object instance annotations in the form of bounding box or/and segmentation mask, which can be used to develop and evaluate object proposal generation algorithms. In this section, we briefly describe related datasets and introduce the evaluation metrics for object proposal generation.

2.7.1 Datasets

KITTI Object Dataset [15] The KITTI dataset for object detection provides a large number of stereo images of various urban scenes, ranging from freeways over rural areas to inner-city scenes with many static and dynamic objects. It consists of 7,481 training images and 7,518 test images. Every training image is associated with object instance-level annotations, represented by bounding boxes. The object classes involve *Cars, Pedestrains* and *Cyclists*. An advantage of the KITTI dataset is that it provides stereo images, making it feasible to estimate the disparity and to reconstruct the scene. Thus, we can extract geometric features based on the reconstructed depth

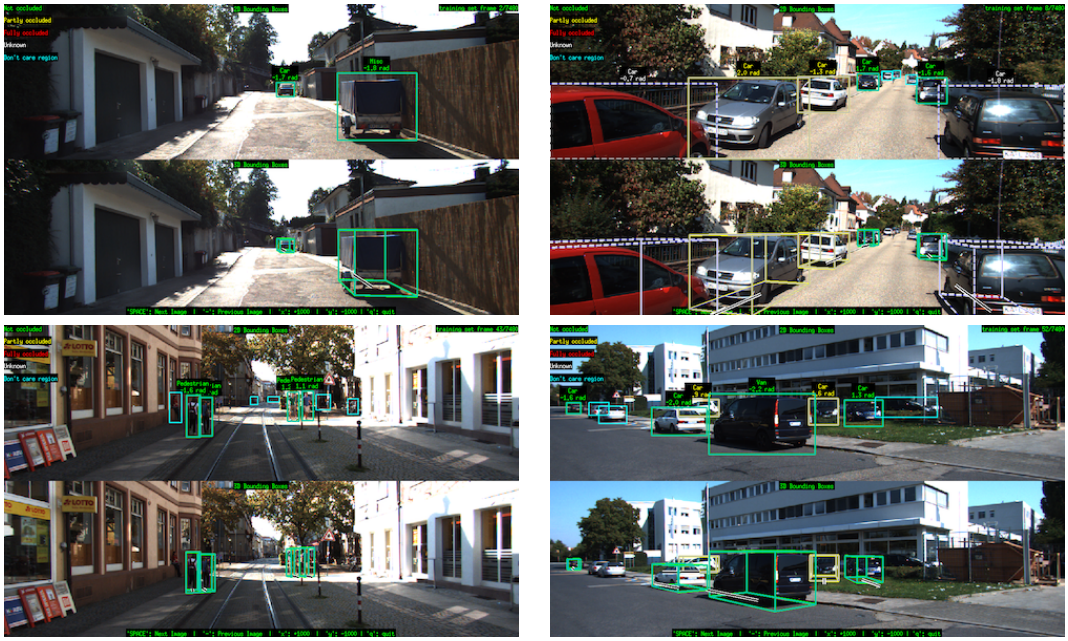


Figure 2.16: Examples for image and ground truth annotations in **KITTI-object** dataset [15].

maps. Figure 2.16 shows several example images and annotations for the KITTI object dataset.

Cityscapes [16] Cityscapes is a newly released large-scale dataset for semantic urban scene understanding. It is comprised of a large diverse set of stereo video sequences recorded on streets from 50 different cities. 5,000 of these images have high quality instance-level annotations for humans *(person and rider)* and vehicles *(car, truck, bus, bicycle, motorbicycle, caravan and trailer)* and they are split into separate training (2,975 images), validation (500 images) and test (1,525 images) sets. In addition to supplying pre-computed disparity maps for images, another obvious strength of this dataset is that it provides segmentation masks for object instances of certain classes. This dataset is very challenging as it is biased towards busy and cluttered scenes where many, often highly occluded, objects occur at various scales. Examples can be seen in Figure 2.17.

PASCAL VOC [17] PASCAL VOC is a well-known benchmark for visual object recognition and detection. It has evolved year by year by adding more images or including new annotations. We mainly use the PASCAL VOC 2012 dataset. As the original dataset provides only a few instance-level segmentation annotations, we actually develop our methods on the Semantic Boundaries Dataset (SBD) [97] (see Figure 2.18). SBD enhances the PASCAL VOC 2012 by providing instance-level semantic segmentation annotations for 11,355 images in it. The SBD inherits the classes from PASCAL VOC, containing 20 categories *(person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant and sofa)*. Besides

Figure 2.17: Examples for the dataset of **Cityscapes** [16]. **Top:** RGB images. **Bottom:** instance-level segmentation ground truth.

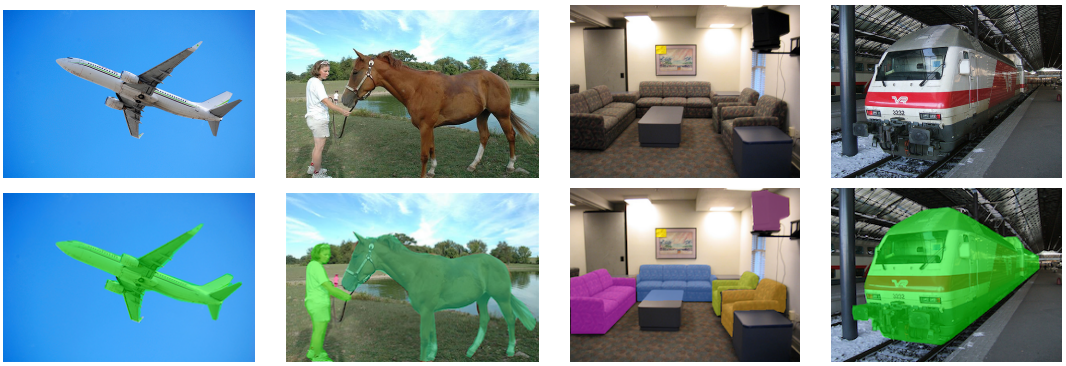


Figure 2.18: Examples for image and ground truth annotations in **PASCAL VOC** dataset [17, 18]. **Top:** RGB images. **Bottom:** instance-level segmentation ground truth.

SBD, another extension of PASCAL VOC is PASCAL-Context dataset [18], which labels 10,103 images of the PASCAL VOC 2010 with pixel-wise accurate segmentation in terms of 520 additional classes, but it does not have instance-level annotations.

MSCOCO [1] MSCOCO is a large-scale dataset for object recognition. It contains 91 common object categories and in total it has 2,500,000 labelled instances in 328,000 images. The main advantage of MSCOCO dataset is that it provides instance-level annotations for a huge number of images. It is a timely dataset in this deep learning era, which big data is vital. It helps to train large-scale deep networks for detecting and segmenting objects. In general practice, 80k images are used as training images, which contain 81 labelled categories, and 40k images for validation. Examples can be seen in Figure 2.19.

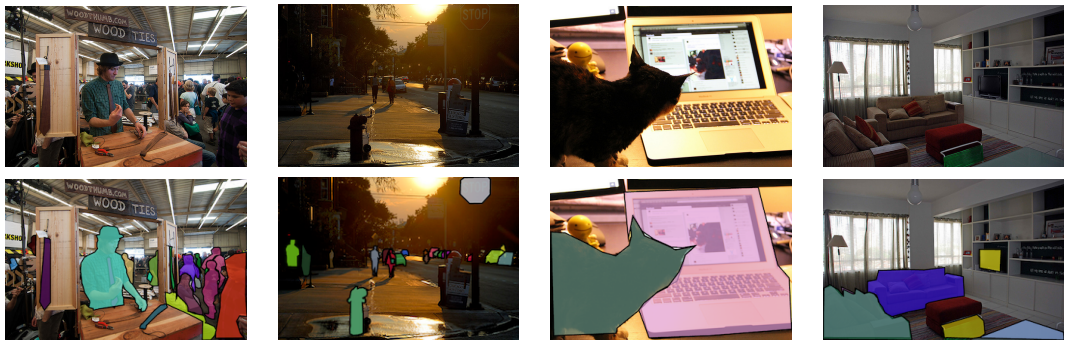


Figure 2.19: Examples for image and ground truth annotations in **MSCOCO** dataset [1]. **Top:** RGB images. **Bottom:** instance-level segmentation ground truth.

2.7.2 Evaluation Metrics

In object proposal generation, performance can be evaluated using several criteria, most of which are computed based on the Intersection over Union (IoU) score. IoU computes the intersection of a proposal and the ground truth divided by the area of their union. This metric can be applied to both bounding box and segment proposals. In the setting of object proposal generation, an IoU score for an object proposal means the highest IoU that this proposal has with all ground truth boxes or segments. Based on the IoU score, we use the following metrics to evaluate our algorithms:

Recall vs. Number of Proposals Given an IoU threshold, *e.g.* 0.5, an object is regarded as recalled if any of proposals has an IoU with this object greater than the threshold. This metric computes the recall when the number of proposals is varied for a fixed IoU threshold.

Recall vs. IoU Thresholds This metric calculates the recall when varying the IoU threshold for a fixed number of proposals. It reflects the quality of proposals in terms of alignment with the ground truth.

Average Recall vs. Number of Proposals Recently, a novel metric for evaluating proposals, the *average recall* (AR), is proposed by Hosang *et al*. [98], which has become a standard metric for evaluating object proposals at present. They show that AR has a strong correlation with the final detection performance. They compute the AR between IoU 0.5 and 1 and report AR versus number of proposals. Given a fixed number of proposals, the AR between IoU 0.5 and 1 can be computed by:

$$AR = 2 \int_{0.5}^{1} recall(o)\ do \tag{2.12}$$

where o is the IoU and $recall(o)$ means the recall vs. IoU thresholds. More recently, in order to be consistent with the evaluation metrics used in the MSCOCO dataset [1], serval studies [12, 99] begin to compute AR between IoU 0.5 and 0.95, instead of between IoU 0.5 and 1, when reporting AR versus number of proposals.

AR Averaged Across All Proposal Counts (AUC) As a supplementary metric, AUC computes the averaged AR across all counts [11, 12].

2.8 Summary

In this chapter, we survey the literature relating to object proposal generation. We first describe those vision tasks that benefit from the object proposal. Then, we mainly review the early stage object proposal generation methods, including bounding box proposal methods and segment proposal approaches. We also analyse the limitations existing in those previous methods, and propose to exploit geometric information, semantic context and feature representation learning in object proposal generation, extending proposal generation to stereo images. At the same time, to convert the stereo images into point clouds for computing geometric features, we simply describe the disparity estimation method used in our work.

Further, we introduce the recent advances of the convolutional neural networks (CNNs) and different ways of using the CNN as a feature extractor for different tasks. The resurgence of CNN greatly promote the progress of object proposal generation. Compared to early stage methods, CNN-based proposal methods that are reviewed in this chapter achieve substantial improvements. However, there is still a large space to boost the quality of object proposals. An alternative approach to producing better object candidates is to refine an initial pool of object proposals by warping them closer to their ground truth. From this perspective, we describe related work on modelling and learning spatial transforms, establishing the basis for our work. Finally, we introduce related object recognition datasets and evaluation metrics for object proposal generation.

Chapter 3

Semantic Context and Depth-aware Object Proposal Generation

3.1 Introduction

Generating object proposals has become a critical step in top-performing object detection systems [32, 8, 65], which helps reduce the search space of detection to a relatively small number of interesting regions [98]. Such reduction improves not only the computational efficiency but also the accuracy of detection methods thanks to much fewer background clutters. Early work of object proposal generation focuses on exploiting local image cues, including object contour [36], edge density [37] and over-segmentation [31, 40, 5]. It usually requires generating thousands of object proposals per image to achieve high recall rate and accurate localization in detection. More recently, learning-based methods have been proposed to refine an initial set of proposals or to directly generate them from images based on deep network features [41, 38, 39, 10]. In addition, 3D shape cues are learned from dense depth images for indoor scenes [100]. These new proposal generation methods generally further improve the quality of object proposals and lead to better object detection and localization performance.

Despite the progress, most of existing proposal generation approaches extract objectness cues from single modality and focus on low- or mid-level features. On the other hand, the spatial locations of object instances need to satisfy certain geometric/physical constraints and have close relations to their neighboring object classes, such as supporting relation and relative size. As such, incorporating geometric and semantic context cues can benefit the proposal generation and further improve their quality.

It has been widely acknowledged that global context plays an important role in object detection and recognition [101]. Several types of contextual information have been explored in the object detection literature, such as scene geometry [102], co-occurring object classes [103], and semantic scene labeling [104]. However, little attention has been paid to exploiting context information in the stage of object proposal generation. A notable exception is the recent work by Chen *et al*. [65], which uses depth context to improve the object proposal generation. However, they focus

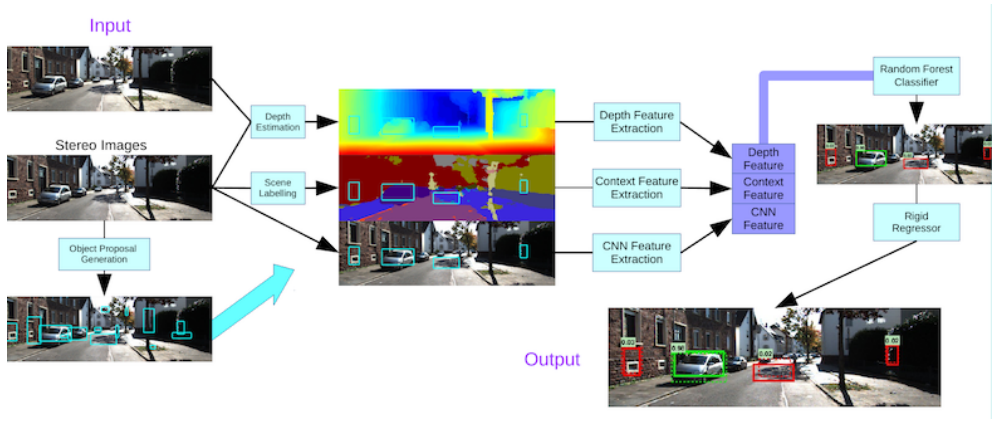


Figure 3.1: Overview of our object proposal generation pipeline. The input are a pair of stereo images and an initial set of proposals. We extract three types of object and context cues, and use them to re-rank the proposals and refine their locations.

on the class-dependent object proposals and use estimated ground plane to reduce their search space, which is restrictive for generic scene understanding.

In this chapter, we propose a novel object proposal generation pipeline, which exploits additional geometric and semantic context cues to improve the recall and localization accuracy of object proposals. To this end, we take a pair images from a stereo camera as input and start from a set of initial object proposals generated from applying the Edgeboxes method [37] to the left image. Our goal is to refine this set of proposals by re-ranking them and fine-tune their spatial locations based on a new set of object and context information.

Specifically, we consider the following three kinds of objectness cues. First, we use the noisy depth computed from the stero images to estimate a set of geometric features on each object candidate; second, we design a semantic context feature to describe the surrounding object class distribution, which is computed from a noisy semantic labeling; finally, we follow the Deepbox method [38] and extract a CNN feature from each object candidate. We then fuse these object and context cues to re-rank the initial object candidates. In particular, based on those features, we train a classifier to predict a new objectness score for each candidate, and regressors to adjust the location of its bounding box. Figure 3.1 illustrates the overview of our approach.

We evaluate our method on the KITTI dataset [15], one of the large-scale publicly available datasets with both stereo images and object annotation. We show that our method improves the quality of the initial object proposals and achieves the state-of-the-art performance. Our main contributions are summarized as follows: 1) We propose a new pipeline for improving object proposals based on additional geometric and semantic context cues; 2) We design a set of geometric and semantic context features that can be efficiently computed (Section 3.2); 3) We systematically evaluate our method on the KITTI dataset and achieve the state-of-the-art recall rate with much fewer proposals (Section 3.3).

3.2 Our Approach

We take as our system input a pair of stereo images and aim to generate a set of high-quality object proposals for its left image. Our approach consists of three stages, as illustrated in Figure 3.1. We first generate a set of initial object proposals in the left image. Given the initial object proposals, we then compute three sets of object and context features for each object proposal, including its geometric properties, the CNN feature and a semantic context feature. Finally, we concatenate these features and train a classifier to re-rank as well as regressors to re-locate those initial candidates. We now introduce the details of each stage of our pipeline, focusing on the feature design and classifier plus regressor training.

3.2.1 Preprocessing

The preprocessing stage generates a set of initial object proposals, dense depth and semantic maps for computing context features in the next stage. For generating the initial object proposals, we choose the Edgeboxes algorithm [37] for its efficiency and good Intersection-Over-Union (IOU) quality. An alterative method to generate better initial proposals is to compute the object proposals considering both the stereo pair and the depth, rather than only the left image of the pair. However, integrating all the raw input is a non-trivial task. Besides, this is not also our research focus. Therefore, we simply employ the Edgeboxes to generate the initial proposals on the left image. We use the disparity estimation method [76] to estimate the dense depth map and convert it into a point cloud representation according to the camera parameters. The semantic map is computed based on the SegNet system [105], although any deep Convnet based method can be used here. The SegNet is pre-trained on the CamVid dataset [106] and generates a pixel-level label map with 12 semantic classes, which are commonly seen in street scenes. We note that no object instance information is available from their outputs.

3.2.2 Object and Context Features

Given each initial object proposal, we compute three types of features to capture its appearance, shape and its geometric context, as well as the semantic context.

CNN Feature For each candidate bounding box, we adopt the CNN feature to encode the object appearance. Specifically, we extract the CNN feature in the same way as in the R-CNN method [32]. We normalize each bounding box into a size of 224×224 and apply the AlexNet [6] network. The network weights are pre-trained on the ImageNet [82] and fine-tuned on the VOC 2012 dataset [107]. We take the output from the layer $fc6$ as our CNN feature, which has 4096 dimensions.

Geometric Feature To incorporate geometric property of the object, we make use of the depth map estimated from the stereo images. We first segment out the subset of the point cloud using the bounding box associated with a proposal. The subset

is used to compute a 12-dimensional feature to describe the object's geometric properties. Specifically, denoting the position of a 3D point as (x, y, z), we consider the following set of features, including *mean x, mean y, mean z, median x, median y and median z* of all points in the bounding box and the *x, y and z* of the center point, as well as the *width, height and depth span* of all points in the box.

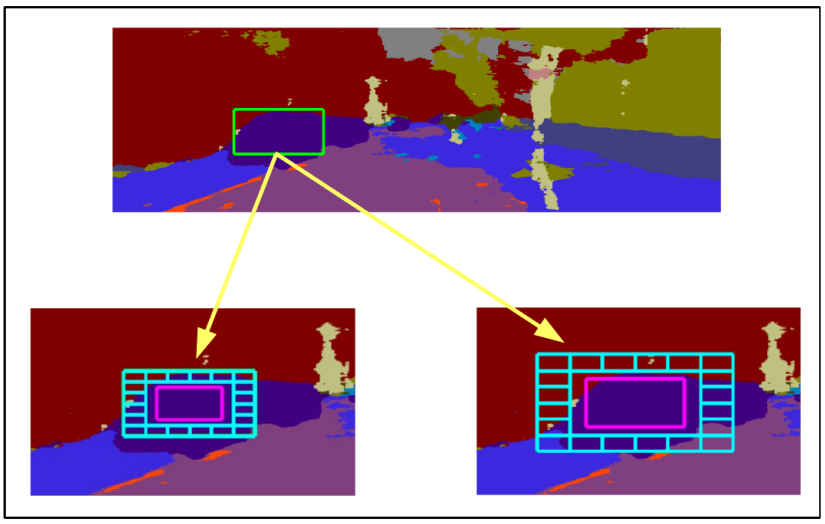


Figure 3.2: The design of semantic context feature, which shows the partition of a bounding box for computing the label histogram. See text for details.

Semantic Context We encode the semantic context of each object proposal by computing a semantic layout feature on the pixel-wise semantic label map. Specifically, each pixel is labeled into 12 classes: *sky, road, road marking, building, pavement, wall/fence, pole, vegetation, car, pedestrian, sign and cyclist*. We split the bounding box into $n \times n$ cells (we use $n = 6$ in our experiment) on the label map. For each of those cells which are next to the boundaries ($4n - 4$ cells in total), we compute a label histogram. Besides that we also compute the label histogram of the inner box whose area is a quarter of the original bounding box. In order to better capture context information, we enlarge the original bounding box by 1.5 times in terms of area and then compute the histograms in the same way as for the original bounding box. Finally, we concatenate these histograms computed from the original and the enlarged bounding box as the semantic context feature. Figure 3.2 shows an example of computing the semantic context feature.

3.2.3 Re-rank Proposals

We concatenate all the features computed from Section 3.2.2 and re-rank all the initial object proposals based on these features. We adopt the random forest (RF) [108] as our classifier for its efficiency during test. To train the random forest classifier, we build our training dataset as follows. We treat the ground-truth bounding boxes and those proposals with ≥ 0.5 IoU overlap with a ground-truth box as positives. Those proposals with ≤ 0.4 IoU overlap with a ground-truth box are labeled as

negatives. We use the held-out validation set to optimize the hyper-parameters in the RF classifier. Our RF classifier consists of 15 trees with a maximal depth of 20 and at least 2 leaf nodes. The RF generates a probability score for each proposal, which is used as the new objectness score.

3.2.4 Bounding Box Regression

Inspired by [32], we learn bounding box regressors to fine-tune the location of each proposal. Note that our regressors are class-agnostic. We represent the bouning box by its center coordinates, its width and height, $\{B_x, B_y, B_w, B_h\}$. The groundtruth box is denoted as $\{G_x, G_y, G_w, G_h\}$. We define the regression targets $\{T_x, T_y, T_w, T_h\}$ for the training pair (B, G) as follows,

$$\begin{aligned} T_x &= (G_x - B_x)/B_w, \quad T_y = (G_y - B_y)/B_h \\ T_w &= \log(G_w/B_w), \quad T_h = \log(G_h/B_h) \end{aligned} \tag{3.1}$$

We learn four linear regressors with the same features as the RF classifier. For each regressor, we estimate the weights β by minimizing the regularized least squares objective:

$$\beta = \arg\min_{\widehat{\beta}} \sum_{i=1}^{N} (T^i - \widehat{\beta}^T f(B^i))^2 + \lambda \parallel \widehat{\beta} \parallel^2, \tag{3.2}$$

where $f(B^i)$ denotes the feature extracted for the bounding box B^i, and λ is the weight for the regularization term. For learning these regressors, we only use those proposals which have ≥ 0.5 IoU overlap with a ground-truth box.

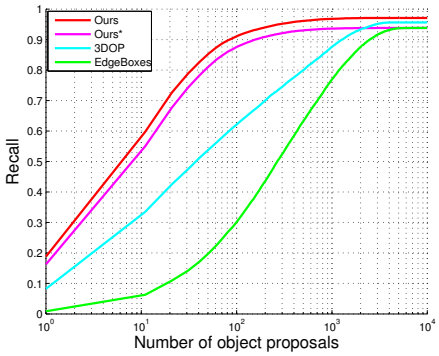
3.3 Experiments

We evaluate our approach on the KITTI object dataset [15], which consists of 7,481 images with bounding box annotations. The object classes consist of *Cars*, *Pedestrains* and *Cyclists*. Similar to the setup in [65], we split the dataset into three subsets: a training set of 3,200 images, a validation set of 512 images and a test set of 3,769 images. We report the results of object proposal generation and object detection task on the test set.

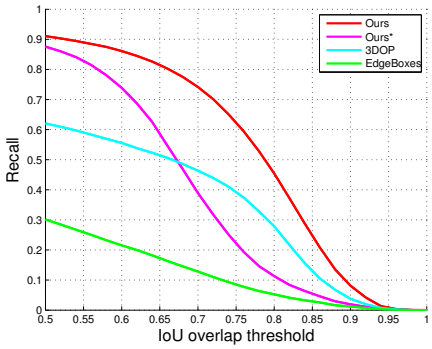
3.3.1 Object Proposal Generation

For object proposal generation, we employ the recall vs. number of proposals and the recall vs. IoU threshold as the evaluation metrics. For the recall vs. the number of proposals, we use 0.5 as the IoU threshold, above which a proposal is treated as recalled [37, 31]. For the recall vs. the IoU, we use top 100 and 1,000 proposals to evaluate the performance.

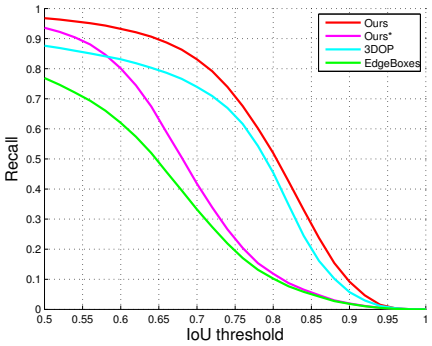
We first compare our algorithm against the baseline method, Edgeboxes-50 [37], and the state-of-the-art, 3DOP [65]. Figure 3.3(a) shows the recall when varying the number of object proposals. We can see that our approach improves the recall rate.



(a) Recall vs. Number of proposals



(b) Recall vs. IoU Threshold (100 proposals)



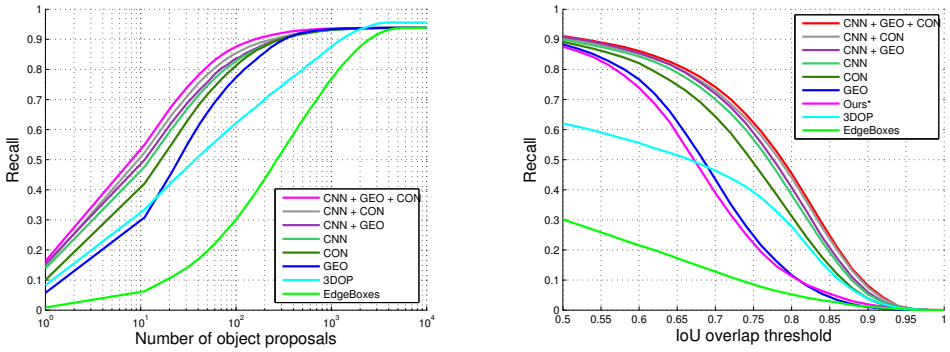
(c) Recall vs. IoU Threshold (1,000 proposals)

Figure 3.3: Comparison of our approach to the baseline and the state-of-the-art (3DOP). 'Ours*' denotes our approach without the bounding box regression. (a): Recall vs. Number of proposals, (b): Recall vs. IoU Threshold (100 proposals) and (c): Recall vs. IoU Threshold (1,000 proposals).

With just 100 proposals, our approach improves the recall rate to a level above 90%, while 3DOP and EdgeBoxes only achieve 63% and 30% respectively. Furthermore, with recall rate 90%, our method uses only one tenth as many proposals as the 3DOP method, which leads to more efficient object detection. We can also see that the bounding box regression further improves the recall rate of our method, as the highest recall rate is further boosted from 94% (the purple curve) to 98% (the red curve).

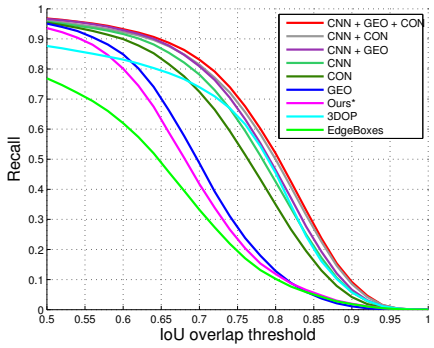
We also show the recall rate when changing the IoU threshold with top 100 and 1,000 proposals in Figure 3.3(b) and 3.3(c). We can see that our approach clearly outperforms the baseline and the state-of-the-art. Interestingly, the bounding box regression improves the proposals location precision obviously. We note that 3DOP uses the object size priors learned for each class, which are unavailable to our method.

Qualitative results are shown in Figure 3.5. It can be seen that our method predicts quite good objectness scores for initial object proposals and relocates the bound-



(a) Effectiveness of features on the object proposals re-ranking

(b) Effectiveness of features on the bounding box regression (100 proposals)



(c) Effectiveness of features on the bounding box regression (1,000 proposals)

Figure 3.4: Ablation study of our features on proposal re-ranking and bounding box regression. (a): Effectiveness of features on the object proposals re-ranking, (b): Effectiveness of features on the bounding box regression (100 proposals) and (c): Effectiveness of features on the bounding box regression (1,000 proposals).

ing boxes much better than initial object proposals.

3.3.2 Ablation Study

To understand the effectiveness of different features, we conduct the ablation study as follows. In the re-ranking stage, we use different groups of features to train the classifier. Figure 3.4(a) shows the recall rate curves with different combinations of our features. We can see that using the geometry features or the semantic context feature alone can improve the recall rate. All the features contribute to the final improvement of recall performance. We also apply the same study to the regression stage and show the results in Figure 3.4(b) and 3.4(c). We can see that the geometry features are not very effective in the bounding box regression, but the context feature is quite powerful. Both studies verify the strength of the CNN feature.

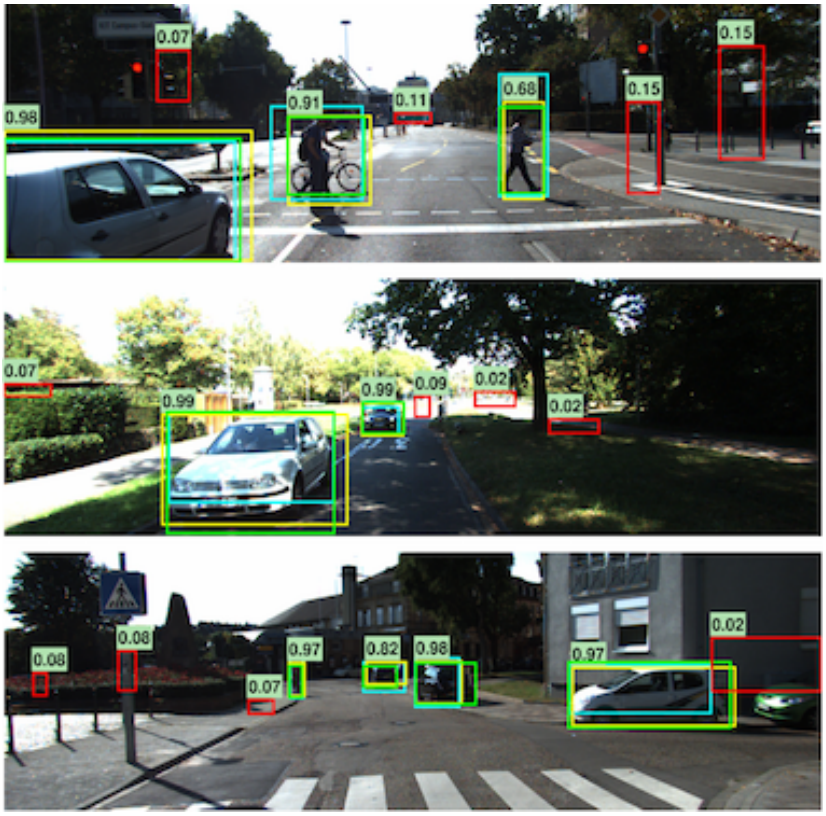


Figure 3.5: Qualitative examples of our object proposals. Green, cyan and yellow bounding boxes are the ground truth, initial proposals and the refined proposals respectively. Red indicates the false positives. Numbers are the new objectness scores.

	Cars			Pedestrians			Cyclists		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
3DOP	45.74	37.79	32.48	51.62	45.57	**41.24**	**29.96**	**22.41**	**21.30**
Ours	**52.39**	**44.88**	**37.33**	**52.21**	**46.45**	41.02	23.51	21.84	20.59

Table 3.1: Average Precision (%) of object detection on the test subset with top 1,000 proposals. We use the class-agnostic version of 3DOP and our approach to generate the proposals respectively. ('Mod' means Moderate.)

3.3.3 Object Detection

To demonstrate the benefit of our proposal generation method, we evaluate the performance of object detection using our proposals. We train a set of object detectors based on a random forest classifier (20 trees with a maximal depth of 25 and at least 3 leaf nodes), which take the same feature set as in Section 3.2.2. We compare the results using our proposals and the proposals from the class-agnostic version of 3DOP as the input to the detectors. Table 3.1 shows the average precision of these two systems. Our proposals perform better than 3DOP's in the majority cases. For the category of cyclist, 3DOP uses the learned 3D size priors, which can help get

more precise proposals, as it can be difficult to discriminate the pedestrians from the cyclists just from appearance.

3.4 Conclusion

In this chapter, we propose a new object proposal generation method for stereo images, which exploits additional geometric and semantic context cues. In addition to the CNN feature of proposals, we design geometric features based on depth map and a semantic context feature computed from pixel-level scene labeling. We train an efficient classifier to re-rank the initial object proposals, and learn a set of bounding box location regressors to fine-tune the position of the re-ranked object proposals. Experiments on the KITTI dataset show that our approach achieves high recall rate with a fraction of the initial proposals and outperforms the state-of-the-art.

Chapter 4

Learning to Generate Object Segment Proposals with Multi-modal Cues

4.1 Introduction

While most work in object proposal generation focus on generating bounding boxes for object detection [4, 40, 36, 37], object segments or region proposals play an important role in semantic segmentation and object segmentation [31, 5]. Compared to bounding box proposals, generating object segment candidates is more challenging, as it entails both object-level localization and pixelwise perceptual grouping. Early work incorporate boundary consistency and smoothness priors through superpixel grouping [40, 5] or MRF-based segmentation [31, 66, 41]. They rely on handcrafted image features to group pixels into region proposals, which are largely limited by the inaccurate over-segmentation processes. More recent approaches use deep ConvNets to learn the feature representation and directly predict class-agnostic object masks [11, 3]. However, such end-to-end learning of a deep network makes it difficult to incorporate additional input data from other sensor modalities, such as depth cues [74, 65]. It may require retraining of the full system using a large dataset with instance-level annotations, which can be expensive and time-consuming.

In this chapter, we consider the problem of generating object segmentation proposals with stereo image inputs. To efficiently incorporate the depth cues computed from the stereo, we take an alternative deep learning approach, and learn an iterative merging process for generating a diverse set of high-quality region proposals. Unlike the previous global approaches, we mainly focus on learning a representation for object-driven perceptual grouping, which is an easier problem due to its local nature and potential to be modeled by a simpler network. More importantly, it enables us to design a late fusion strategy to incorporate the noisy depth cues into grouping without retraining the full deep network pipeline.

Specifically, our method consists of two stages. We start from an initial segmentation hierarchy of the left image and sequentially merge neighboring regions in each level of the hierarchy based on affinity scores predicted by a learned similarity net-

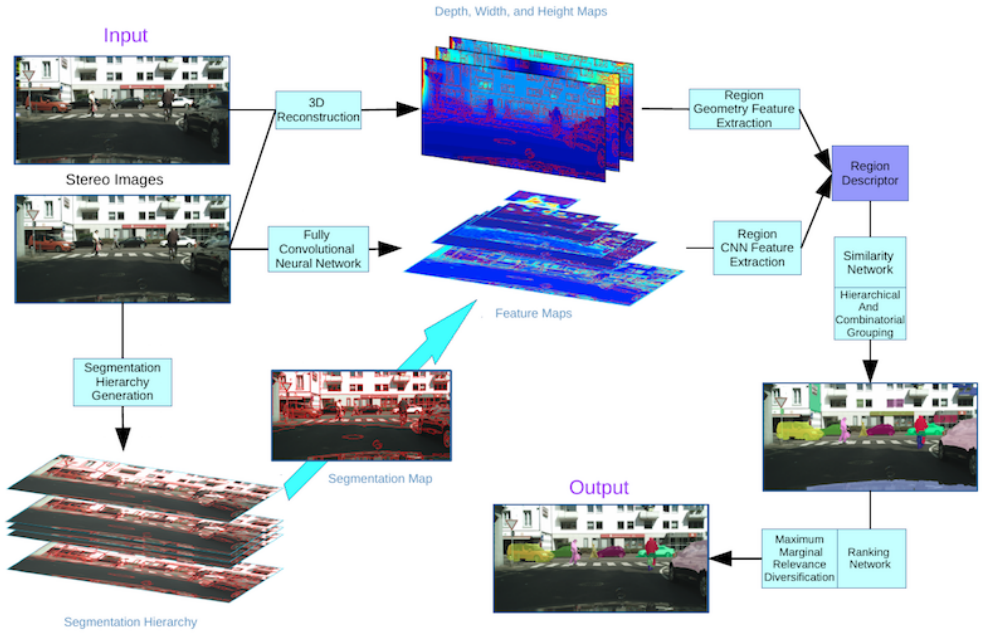


Figure 4.1: **Overview:** Our system takes as input a pair of stereo images. We first generate a segmentation hierarchy, compute the convolutional feature maps and reconstruct the 3D scene. Then, we extract descriptors for regions in the segmentation hierarchy. Next, we iteratively merge adjacent regions based on their affinity score predicted by a similarity network to generate object proposals. Finally, we rank these object proposals through a ranking network and diversify the ranking.

work. This merging process generates new hierarchies of image segments, which is used to produce a pool of regional proposals by taking single, pair, triple and 4-tuple neighboring segments from the hierarchies. We then learn a ranking network to predict the objectness score of each region proposal. Our similarity and ranking network use a combination of learned deep features for appearance and designed geometric features for depth cue. While the similarity network predicts how likely two regions belong to the same object instance or the same background class, the ranking network estimates the overlap ratio with respect to the ground truth for each candidate region.

We evaluate our algorithm on the Cityscapes dataset [16] with comparisons to Selective Search [40] baseline and several stat-of-the-art methods, including Multi-scale Combinatorial Grouping (MCG) [5] and Geodesic Object Proposals (GOP) [42]. Our results show that we achieve improvements over these methods The main contributions of our work are three folds: first, we propose a deep learning approach to the multi-modal object segmentation proposal generation; second, we design an alternative method to produce region proposals with a learned merging network and ranking network; and finally, our method achieves superior performance to the strong baselines on the challenging Cityscapes dataset.

4.2 Our Method

We aim to generate a set of object segment proposals and their objectness scores from a pair of stereo images. To this end, we design a segment proposal generation pipeline that learns to fuse multi-modal cues and to merge oversegmentation into object candidates. Figure 4.1 illustrates an overview of our approach. We first estimate a dense depth map of the scene using the stereo images, and build a segmentation hierarchy of the left image. Given the initial segmentation hierarchy, we represent all the regions in the hierarchy using convolutional and depth features. We then train a neural network to predict the affinity of neighboring regions, and rebuild multiple segmentation hierarchies by incrementally merging adjacent regions from all the levels based on the learned similarity. From the new segmentation hierarchies, we extract region singletons, pairs, triplets and 4-tuples as object segment proposals. Finally, we rank these object proposals through a learned ranking network and diversify the ranking based on Maximum Marginal Relevance measures [31]. We now describe each stage of our pipeline in detail.

4.2.1 Initial Segmentation Hierarchy Generation

The first step of our method constructs an initial segmentation hierarchy of the left image. To generate the segmentation hierarchy, we use the Structured Edge Detection [109] on the left image to obtain an edge map for its efficiency and accuracy. It would be better to compute the segmentation hierarchy using both images in this stereo pair. But there were no such existing methods off the shelf then. Also, this is not our focus, so for simplicity we just use algorithms handy to first generate the edge map from the left image. An Ultrametric Contour Map (UCM) [5] is generated based on the estimated edge probability map. Then we threshold the UCM at five different levels to create the segmentation hierarchy. The thresholds are chosen such that the numbers of regions from the base level to the top level are roughly 1024, 768, 512, 384 and 256, respectively. For every region, we also record its child regions in the hierarchy, which enables efficient propagation of region descriptors from the base level to higher levels in the hierarchy.

4.2.2 Multi-modal Region Representation

For each region in the segmentation hierarchy, we extract two types of features to capture its appearance and 3D geometric properties. We take an efficient bottom-up approach to compute the region features at all the hierarchy levels. We only need to calculate those features explicitly for the base level regions and use max-pooling or weighted average-pooling to obtain features of higher level regions recursively.

4.2.2.1 Appearance Features

We extract a set of rich deep features to encode the appearance of a region. We first feed the left image into a Fully Convolutional Network (FCN) [80] to generate

multiple layers of feature maps for the entire image. We choose the FCN-8s model initialized by VGG-16 [85] trained on PASCAL-Context dataset [18] for the scene labeling task due to its superior performance and diverse set of 59 semantic classes, including *sky, ground, grass, building, road, person,bicycle and car*, *etc*. The feature map outputs from $pool1, pool2, pool3, pool4, pool5$ *and* $fc7$ *layers* are used as our representation, inspired by the "Hypercolumns" concept proposed by Hariharan *et al*. [7].

Given the feature maps, we compute the appearance features of a region by masking and max-pooling. As the feature maps of different layers are not of the same size, the deep features of a region cannot be directly masked out from these maps. A straightforward way to solve this problem is to upsample the feature maps to the same size as the image [7]. However, due to high dimensionality and varying sizes of the feature maps, *e.g*. the output from $fc7$ *layer* has 4,096 dimensions and a very small size (17×33 in our case), such upsampling is very time-consuming and memory-costly.

To tackle this issue, we adopt the convolutional feature masking technique proposed by Dai *et al*. [9]. Specifically, we first compute the receptive field for every neuron activation in each layer according to the receptive field geometry [110]. Then we project each neuron activation onto the image plane, which is located at the center of its receptive field. We define a "domain of influence" for a neuron on the image plane, which has the same center as its receptive field and a smaller width (or height) that equals to the distance between neighboring receptive field centers. For example, the neuron at location $(1, 1)$ in $pool5$ *layer* may have a square "domain of influence" with its center at $(16, 16)$ and its side length as 32 in the image domain. If over 50% of the "domain of influence" of a neuron is covered by a region mask, we label this neuron activation as active for this region and it will be included in the calculation of region feature. By this labeling process, we project the base level region masks in the image plane onto the feature maps and then we do max-pooling in the projected masks on the feature maps to extract regions' deep features. Figure 4.2 (left) shows an example of our feature computation process. This generates a $5,568$-dimensional feature to encode the region's appearance. Note that when computing the deep features of regions in higher levels of the hierarchy, we only need to do max-pooling among their child region features.

4.2.2.2 3D Geometric Features

To encode geometric properties of a region, we extract two sets of 3D geometric features. We first estimate the dense depth map using the method [76] and convert it into a point cloud representation in the camera coordinate system according to the provided camera parameters.

Given the point cloud and a base level region mask, we segment out the subset of the point cloud using the mask. The subset is used to compute two sets of features to describe the region's geometric properties. Denoting the position of a 3D point as (x, y, z), we first compute the center of the region as one set of features, including $mean\ x, mean\ y,$ *and mean* z. Another set of features describe the spatial distribution

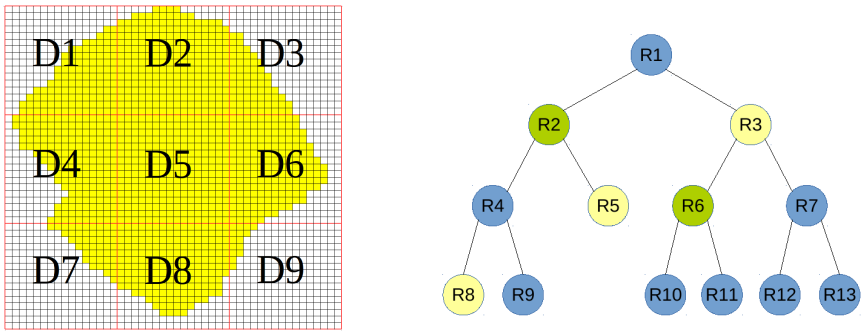


Figure 4.2: **Left:** Illustration of "domain of influence" and feature masking. D1∼D9 red rectangles are the domains of influence of activations A1∼A9 in *pool5 layer*. The yellow mask is a region and only A2, A4, A5, A6 and A8 are activated by this region, as over half of their domains of influence are overlapped by this region. **Right:** Illustration of combinatorial grouping. Singletons: R1∼R13. Pairs: (R2,R6). Triplets: (R3,R5,R8).

of the point cloud, consisting of three histograms, one for each dimension of the point cloud. Specifically, for the width, height and depth dimension, we evenly divide the spatial ranges $[-50m, 40m]$, $[-40m, 3m]$ and $[1m, 100m]$ into 256 *bins*, 128 *bins* and 256 *bins* in the log space, respectively. These spatial ranges are obtained from the statistics of the point clouds in the training set. The spatial histograms are computed based on these bins and then normalised by their L_1 norm. The two sets of features are concatenated to form a 643-dimensional feature G_{r_i} to encode the region's 3D geometric properties. The geometric features of higher levels can be efficiently computed through the hierarchy by weighted average-pooling of child region features as follows,

$$G_{r_{parent}} = \frac{\sum_{r_i \in children} G_{r_i} \times area(r_i)}{\sum_{r_i \in children} area(r_i)} \,. \tag{4.1}$$

4.2.3 Similarity Network

Given the segmentation hierarchy, we expect to learn a merging process that generates a high-quality object candidate set from the initial over-segmentation. To achieve this, we design and train a neural network to compute the affinity between two adjacent regions and use the network to merge region pairs recursively. Unlike the manually designed similarity scores used in [40, 70], our network enables us to learn a more effective merging criterion in the multi-modal space.

4.2.3.1 Network Architecture

Our similarity network takes a concatenation of feature descriptors from two adjacent regions as input and consists of three fully-connected layers. Each layer has 512 neurons and uses ReLU as activation function except the last layer. We add the dropout layer after the first two layers to prevent overfitting. The output is the affinity

score between two input regions in the range of $[0, 1]$ and indicates how likely two regions belong to the same object. We use the MatConvNet [110] to implement our networks in this work.

4.2.3.2 Network Training

We obtain the training examples from the initial segmentation hierarchy. As we are learning a similarity network for object proposal generation, we expect that the network is able to output a high similarity score for two regions from the same object instance or the same background class, and to output a low score for two regions from different object instances. This network can be viewed as re-weighing the boundary strength between regions in the original UCM.

We formulate the network training as a binary classification problem. Each positive example is a pair of neighbouring regions overlapped with the same object instance and both regions have an overlap score larger than a threshold $t_{p1} = 0.7$. Here we define the overlap score as the intersection of a region and an object instance divided by the area of that region. We also take pairs overlapped with the same background class. In particular, we hope that regions belonging to the same background class around the object instance can merge together so that they do not interfere with the grouping of those regions from this object instance. To balance the positive examples from the object instance and from the background classes, we keep the proportion between them roughly at $1 : 2$.

For negative examples, we first take pairs of neighbouring regions in which one has an overlap with the object instance higher than $t_{p1} = 0.7$ while the other overlaps with the same object instance less than $t_n = 0.6$. Similar to the positive examples, we also include adjacent background region pairs which satisfy the same overlapping condition. We keep the proportion of negative examples from the object instance and from the background classes at about $1 : 1$.

To mimic the process of grouping at test time, we scan regions from all levels of the segmentation hierarchy and obtain about $4,120,000$ positive and $3,370,000$ negative training examples. As there are two ways to concatenate features from two adjacent regions, we use both orders in the network training and the total number of training samples is doubled.

We train the similarity network to minimize the *log loss* using stochastic gradient descent with a batch size of $2,000$ examples, momentum of 0.9, weight decay of 0.0005 and for 15 epochs. The learning rate we use for each epoch changes from 10^{-3} *to* 10^{-6} evenly in the log space.

4.2.4 Hierarchical and Combinatorial Grouping

Given the region features in every level of the segmentation hierarchy and a learned similarity network, we generate a set of object proposals by a hierarchical and combinatorial grouping process.

4.2.4.1 Hierarchical Grouping

We start from the initial regions in a single level of the segmentation hierarchy and re-group them by applying the similarity network. Specifically, we first compute the affinities between all adjacent regions via forwarding the feature descriptor of neighbouring regions through the similarity network. Then two most similar regions are merged into a new region and the descriptor for this new region is computed. This can be easily done by max-pooling (for appearance feature) or weighted average-pooling (for geometric feature) as described in Section 4.2.2. Next the affinities between this new region and its neighbours inherited from its child regions are updated using the similarity network. This merging process is repeated until the whole image becomes a single region. We apply this hierarchical grouping procedure to all five levels of the initial segmentation hierarchy so as to generate a variety of complementary segmentation trees, and take all single regions (region singletons) in the five new segmentation hierarchies as our initial set of object proposals.

4.2.4.2 Combinatorial Grouping

Selecting only the region singletons in the segmentation hierarchies, however, is insufficient to generate a high quality pool of object proposals. We follow a combinatorial grouping procedure similar to [5] to generate a larger object proposal set. In particular, we empirically select 10,000 region pairs, 10,000 region triplets and 5,000 region 4-tuples from every newly generated segmentation hierarchy to expand our object proposal pool, which performs well in our experiments. Specifically, as we can infer exactly which regions are neighbours and who are their child regions or parent regions with the representation of the segmentation tree, we compute the region neighbours from the top of the tree to a certain depth and then from this list we can easily select region pairs, region triplets and region 4-tuples. For more details, please refer to [5]. Figure 4.2 (right) shows an simple example of region singletons, pair and triplet in the segmentation hierarchy. We perform Non-Maximal Suppression (NMS) afterwards, which significantly reduces the number of candidates, since those region pairs, triplets and 4-tuples from the same segmentation hierarchy are heavily overlapped. The final pool of object proposals contains less than 10,000 proposals per image on average.

4.2.5 Ranking Network

In the final step, we estimate the quality of each object proposal, or its objectness score. This allows us to obtain good trade-off between the number and the quality of object proposals under different settings. We achieve this by training a ranking neural network to predict the IoU of each object proposal with the matched ground truth as in [31].

4.2.5.1 Network Architecture

Our ranking network is a regression network, which has a similar architecture to the similarity network except the input and output layer. It also consists of three fully-connected layers and each layer has 512 neurons. The input is the feature descriptor of a single object proposal, which can be computed efficiently as follows. Proposals defined by region singletons have their descriptors precomputed during the merging process. For those proposals formed by region pair, triplet or 4-tuple, their descriptors can be computed using the same max-pooling or average-pooling method described before. The output layer of the network is a linear layer that predicts the IoU between the input proposal and its corresponding ground truth. We minimize the the mean squared loss during network training. In the training stage, we also add a dropout layer after the first two layers to prevent overfitting.

4.2.5.2 Network Training

We build the training dataset by choosing four types of training examples. The first type includes all the ground truths and the corresponding target IoUs are therefore 1.0. The remaining training examples come from the object proposals generated on the training set. We split these object proposals into three categories according to their IoU with the ground truth:$IoU >= 0.5$, $0 < IoU < 0.5$ and $IoU = 0$. For the first category, we take all proposals in this group as training examples and denote its size as N. As to the latter two categories, we randomly select $3N$ and $3N$ examples from their pools respectively, which relatively balances the training dataset. Finally, we obtain about 5,000,000 training examples in total.

We train the ranking network using stochastic gradient descent with a batch size of $2,000$ examples, momentum of 0.9, weight decay of 0.0005 and for 10 epochs. The learning rate we use for each epoch changes from 10^{-2} *to* 10^{-5} evenly in the log space.

4.2.5.3 Diversifying the Ranking

After assigning every proposal a ranking score, we diversify the ranking to reduce redundancy. Following [31], we achieve this based on Maximum Marginal Relevance measure, which is used to remove redundant object proposals. We apply the same re-ranking procedure as in [31] to lower the rank of the segment proposals that heavily overlap with higher-ranked proposals.

4.3 Experiments

In this section, we evaluate our multi-modal object proposal generation approach on the publicly available Cityscapes dataset [16]. To the best of our knowledge, Cityscapes is the only public dataset with stereo images and object instance segmentation ground truth, which are required by our method for quantitative evaluation.

Figure 4.3: Illustration of the Cityscapes dataset.**Top:** RGB images. **Bottom:** instance-level ground truth.

4.3.1 Dataset

Cityscapes [16] is a newly released large-scale dataset for semantic urban scene understanding. It is comprised of a large diverse set of stereo video sequences recorded on streets from 50 different cities. 5,000 of these images have high-quality instance-level annotations for humans and vehicles and they are split into separate training (2,975 images), validation (500 images) and test (1,525 images) sets. This dataset is challenging as it is biased towards busy and cluttered scenes where many, often highly occluded, objects occur at various scales. Figure 4.3 shows again some examples.

In our experiments, we further split the training set into two subsets: one for training (2,614 images) and the other for validation (361 images taken at Tubingen, Ulm and Zurich). We use their validation set (500 images) to evaluate the approaches, as the ground truth of the test set is withheld and their evaluation server does not provide results on proposal generation. The original image size is 1024×2048, which is too large to feed into the GPU memory when forwarding the image through the FCN-8s. So we downscale the original image by a factor of 4 into 512×1024. The dataset only provides instance-level annotations for humans (*person and rider*) and vehicles (*car*, *truck*, *bus*, *bicycle*, *motorbicycle*, *caravan and trailer*), which are considered as object proposal ground truth in our experiments.

4.3.2 Evaluation Measures

We employ the recall vs. number of proposals with a fixed IoU threshold and the average recall (AR) as the evaluation metrics. As discussed by Hosang *et al*. in their work [98], AR has been shown to have a strong correlation with the final detection

Method	AR@100	AR@1000	AR@5000	AR@N	AUC	AUC^S	AUC^M	AUC^L
SeSe-Fast	-	-	-	0.122	0.106	0.052	0.206	0.358
SeSe-Quality-10k	-	-	-	0.137	0.108	0.047	0.221	0.402
SeSe-Quality-60k	-	-	-	**0.174**	0.145	**0.077**	0.278	0.451
MCG	0.045	0.087	0.113	0.115	0.107	0.041	0.229	0.432
GOP(200,15)	0.032	0.059	0.065	0.065	0.063	0.001	0.169	0.406
GOP(140,4)	0.032	0.056	0.056	0.056	0.055	0.001	0.151	0.344
Ours-noDepth-Seg	0.086	0.140	0.165	0.166	0.159	0.069	0.335	0.559
Ours-Depth-Seg	**0.099**	**0.150**	**0.165**	0.166	**0.160**	0.070	**0.337**	**0.566**

Table 4.1: AR at different number of proposals(100, 1,000, 5,000 and total number of proposals(N)), overall AUC (AR averaged across all proposal counts) and also AUC at different scales (small, medium and large objects denoted by superscripts S,M and L).

performance. In our experiments, we compute the AR between IoU 0.5 to 1 and report AR vs. number of proposals.

4.3.3 Baseline and State-of-the-Art

As we focus on object segmentation proposals generation, we mainly compare our approach (Ours-Depth-Seg) against two widely-used top-performing segmentation proposal generation methods: MCG [5] and SelectiveSearch [40], as well as our approach without geometric features (Ours-NoDetph-Seg). In addition, we compare to the more recent 'Geodesic Object Proposals' (GOP) method [42] (GOP(200,15) and GOP(140,4)), which has publicly available code.

We use the default parameters in MCG to generate the proposals. For Selective Search, we adopt the parameters used in RCNN [32], and keep the segmentation proposals instead of bounding boxes. The "Quality" version of Selective search (SeSe-Quality-60k) uses four different initial segmentations, five color spaces and four similarity functions to diversify object proposals and over 60,000 proposals are generated per image on average. To make a fair comparison, we randomly select 10,000 proposals (SeSe-Quality-10k) from the SeSe-Quatlity-60k and evaluate their quality. We repeat this for 5 times and take the average results as their performance. The "Fast" version (SeSe-fast) uses only two different initial segmentations, two color spaces and two similarity functions for diversification and about 12,000 proposals on average are generated per image.

Furthermore, in order to demonstrate that our method can also generate high-quality bounding box proposals, we conduct experiments to compare with the Edge-Boxes [37]. We use the tightest boxes enclosing our segmentation proposals as the output to evaluate our method.

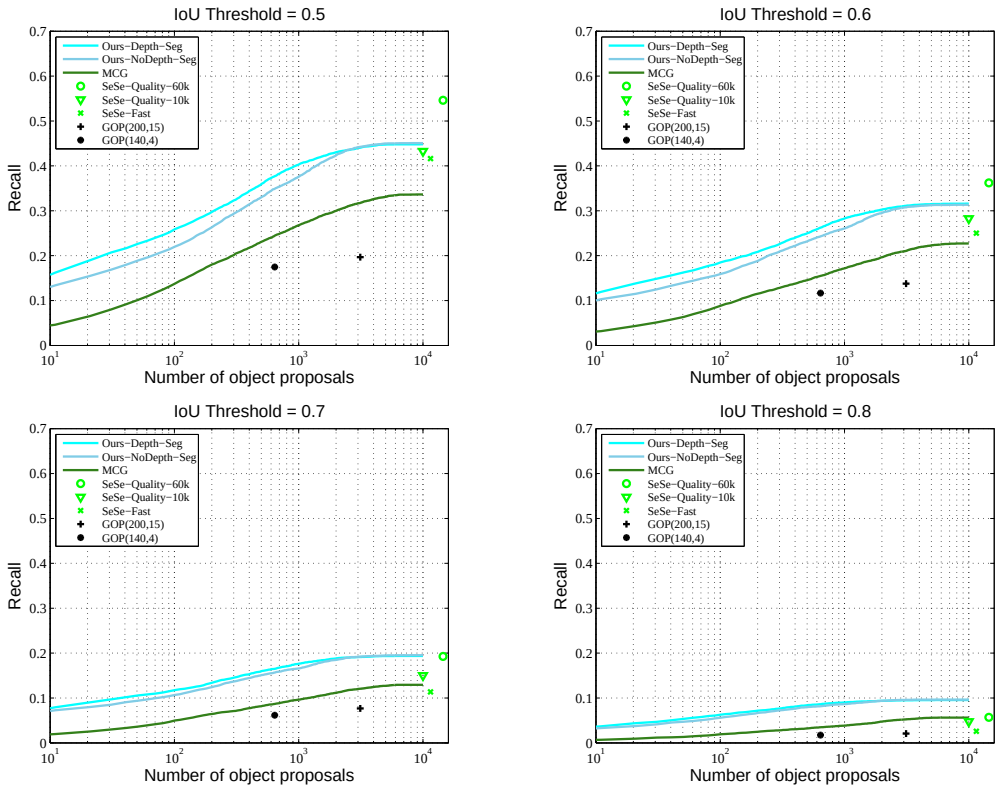
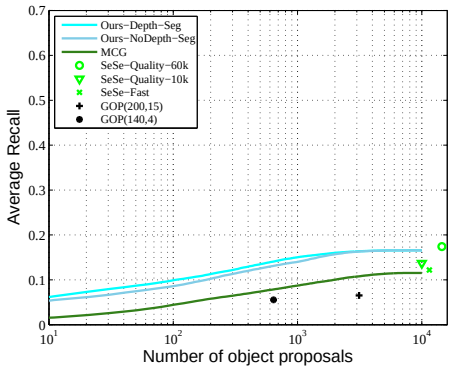


Figure 4.4: Recall vs. number of proposals under different IoU thresholds.

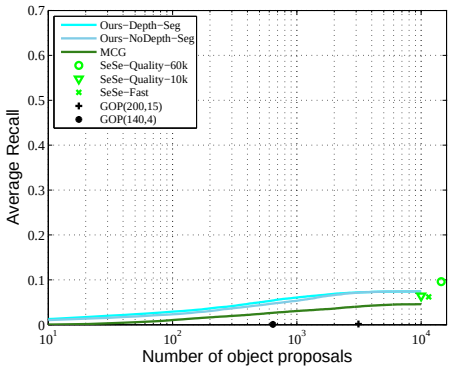
4.3.4 Segmentation Results

Figure 4.4 shows the recall rate when varying the number of object proposals under different IoU thresholds. We can see that our approach constantly outperforms MCG, SeSe-Quality-10k, SeSe-Fast and GOP. The recall of our approach attains 44.8% at about 5,000 proposals when IoU threshold is 0.5, while MCG attains 33.6%, SeSe-Fast 41.7%, and SeSe-Quality-10k 44.0%. The performance of both versions of GOP is much lower than the above methods. With 1,000 proposals and IoU threshold as 0.5, the recall of our approach is above 40.0% while MCG just gets 26.8%. When the IoU threshold increases, we can see that the performance of Selective Search drops much faster than Ours and MCG, and particularly when the IoU threshold equals to 0.7, our method has a similar recall as SeSe-Quality-60k. This indicates that the quality of our proposals is better than Selective Search.

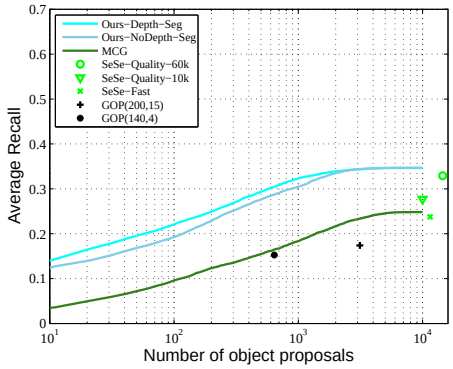
On the other hand, the performance of our approach using geometry information is always better than that without geometry information. Surprisingly, the upper bound of recall is not boosted by geometry information. This might be due to the noisy depth cues computed from the stereo images and that the geometric feature we manually designed is relatively weak. However, the ranking of proposals indeed benefits from the additional geometry information, as geometry information like the 3D height of a region is a good indicator of the objectness in street scenes.



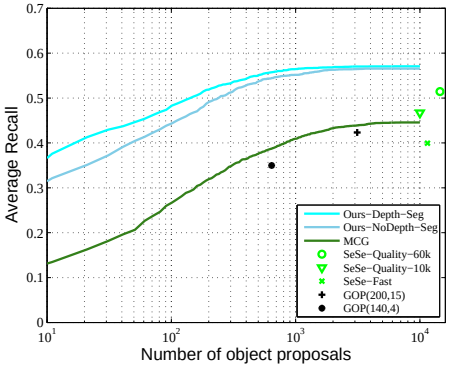
(a) AR vs. Number of proposals (Overall)



(b) AR vs. Number of proposals (Small objects)



(c) AR vs. Number of proposals (Medium objects)



(d) AR vs. Number of proposals (Large objects)

Figure 4.5: AR vs. number of object proposals: (a) overall, (b) small objects, (c) medium objects and (d) large objects.

Figure 4.5(a)describes the overall AR when changing the number of object proposals. It shows that our approach is better than MCG, SeSe-Fast, SeSe-Quality-10k and GOP, but slightly worse than SeSe-Quality-60k which uses much more proposals. With 1,000 proposals, our method achieves an AR of 15.0%, while MCG just 8.7% and this number is consistent with the performance of instance segmentation task reported by Cordts *et al*. [16] who use MCG object proposals in their experiments.

Following [11], we also report the AR vs. the number of object proposals at different object scales, as the size of object in Cityscapes dataset varies in a quite wide range. We split the ground truth into three sets according to object pixel area a: small ($a < 32^2$), medium ($32^2 \leq a \leq 96^2$) and large ($a \geq 96^2$). Figure 4.5 describes the performance at each scale. All methods perform poorly on small objects (Figure 4.5(b)), which leads to the low overall AR. By contrast, when it comes to the categories of medium (Figure 4.5(c)) and large (Figure 4.5(d)) object, the AR by all approaches has a considerable increase and our method performs substantially better than MCG, SeSe-Fast, SeSe-Quality-10k and GOP, and also slightly better than SeSe-Quality-60k.

More detailed quantitative results are shown in Table 4.1, which reports the AR at

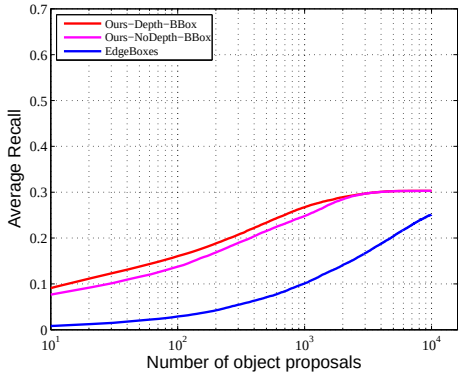


Figure 4.6: AR vs. number of object proposals for Bounding Box proposals.

selected proposal numbers and the averaged overall AR across all proposal numbers (AUC), as well as AUC at different object scales. Finally, examples of generated proposals with the highest IoU to the ground truth on selected images are shown in Figure 4.7.

4.3.5 Bounding Box Results

We also compare our method against bounding box proposal generation method, the EdgeBoxes [37], using the metric of AR. Figure 4.6 shows that our approach generate much better bounding box proposals than the EdgeBoxes. With 1,000 proposals, our approach gets an AR of 27.3%, which is over 2.5× higher than the EdgeBoxes' (10.5%). The upper bound of our method (31.2%) is also much higher than the EdgeBoxes's (25.0%).

4.4 Conclusion

In this chapter, we propose a learning-based object segment proposal generation method for stereo images, which exploits both deep features and the depth cue. We extract features from convolutional feature maps and geometry maps to describe a region. We learn a similarity network to estimate the affinity between two adjacent regions, sequentially merge regions from a segmentation hierarchy based on the affinity to generate object proposals and learn a ranking network to predict the objectness of a proposal. Experiments on the Cityscapes dataset show that our approach achieves much better average recall than the state-of-the-art and depth cue can improve the ranking of proposals.

Figure 4.7: Qualitative examples of our object proposals.**Left:** Ground truth. **Right:** Our best proposals.

Chapter 5

Learning Spatial Transforms for Refining Object Segment Proposals

5.1 Introduction

Recent years, instance-level semantic segmentation, which jointly detects and segments all objects in an image, has attracted much attention [58, 9, 111]. As in most modern object detection systems [8, 10], a critical step in object segmentation is to generate generic object segment proposals for its downstream classification and/or global reasoning [3, 11, 12].

With the prevalence of convolutional neural networks (CNN), more recent approaches to generating object segment proposals learn deep networks to produce binary masks from the image directly, including DeepMask [11], SharpMask [12] and Multistage networks [3]. These CNN-based methods significantly improve the performance of object segment proposal generation, showing impressive results. Nevertheless, learning such a direct mapping from images to segments has shown to be challenging, which usually produces object masks lacking good boundary alignment and requires post-processing to improve their quality.

An alternative approach to generating better object proposals is to refine an initial set of object segments produced by existing methods [12, 111]. Such a strategy enables us to use the initial segment as a starting point and learn additional feature representations for improving the mask accuracy. Hence it is more flexible than the early group-and-rank methods [31, 5]. In addition, as it aims to minimize the residual error between the initial segments and the ground truth, the problem of refinement is conceptually simpler than solving the original image-to-mask mapping task. In essence, it learns a transformation that moves the initial mask predictions 'closer' to the target object segments.

In this chapter, we propose an efficient object segment refinement method that learns spatial transforms to improve the pixel-level accuracy of the object proposals. In contrast to the prior approaches that build a refinement network to predict pixelwise masks [12], our method takes both image and initial object masks as input, and predicts a spatial affine transformation in 2D image plane for each mask, which is then used to warp the corresponding mask into a more accurate object segment

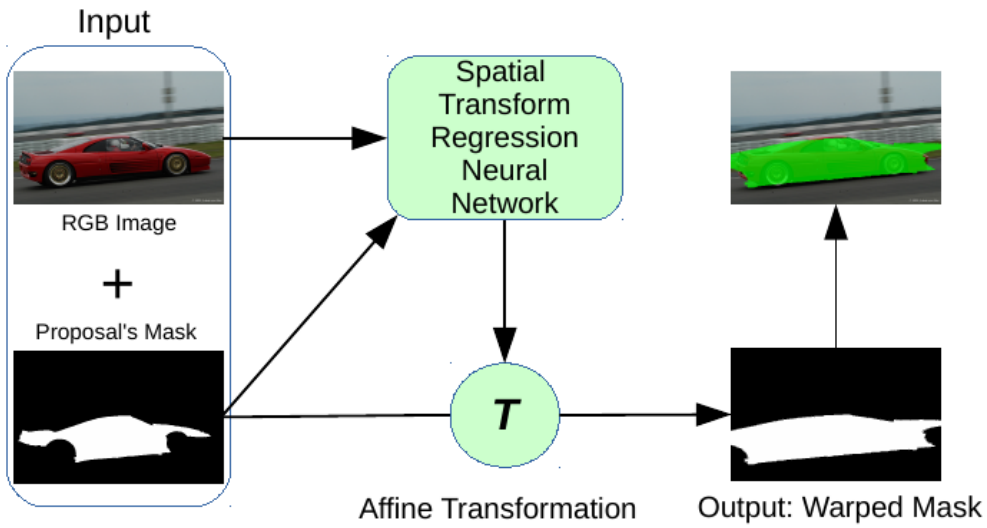


Figure 5.1: Overview of our segment proposal refinement pipeline. We propose to learn a regression network to warp initial segment candidates towards the groundtruth objects.

candidate. Figure 5.1 illustrates an overview of our approach.

Specifically, we formulate the segment refinement as a regression problem, and build a deep network to predict the 2D affine transformation required for improving the mask accuracy. Given the input image, we first extract a hypercolumn feature representation [7] to represent the multiscale image cues. On these feature maps, we design a novel mask pooling scheme that incorporates cues from both an initial object segment and its spatial context. The pooled features are fed into a four-layer neural network, which outputs affine transformation parameters for warping the object mask. To train the regression network, we precompute the affine transformations from the initial object masks to their corresponding groundtruth masks based on nonrigid registration [94], which are used as our regression targets.

We evaluate our approach extensively on two publicly available datasets with object instance segmentation ground truth, the Cityscapes [16] dataset and the PASCAL VOC dataset [17, 97]. Our refinement network is applied to three different sets of initial object segments generated from MCG, DeepMask and SharpMask respectively, and achieves sizeable improvements in the average recall rate across all the experimental settings.

The contributions of our work are three folds: First, we propose a novel refinement method that learns spatial transforms for improving the quality of object segment proposals. Second, we design and train an efficient deep network to predict the instance-level affine transformations based on hypercolumn feature and mask pooling. Finally, our experimental evaluation shows consistent improvements over several state-of-the-art methods on challenging benchmarks. The main strengths of our approach lie in its *generality*, as it can be applied to any initial object segment

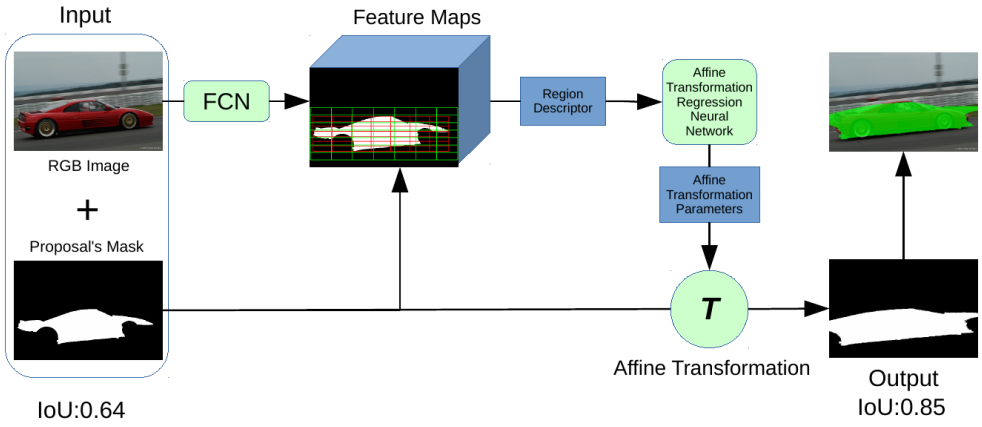


Figure 5.2: Model structure of our approach. Our system takes as input an image and initial segment proposals. It first extracts deep features to describe a segment and feeds the descriptor into a learned regression network to estimate an affine transformation. We then apply the affine transformation to the segment's mask to obtain the warped mask.

proposals; and its *simplicity*, as we only need to predict a spatial transform in a low-dimensional space.

5.2 Our Approach

We aim to generate a set of high-quality object segment proposals for instance-level semantic scene understanding. To this end, we adopt a refinement strategy to improve the object mask accuracy of any initial segment candidate pool generated from existing methods. Our system takes as input an image and the binary masks of its segment proposals, and produces a transformed object mask for each initial segment proposal.

$$T : m_i \mapsto m_g \tag{5.1}$$

To achieve this, we design a deep neural network that predicts an affine transformation for each input segment candidate. Equation 5.1 describes this idea, where m_i is the mask of an iniitial segment proposal and m_g is the groundtruth mask. Our goal is to predict the underlying affine transformation T that can warp m_i close to m_g. In particular, we propose a novel mask feature pooling scheme, which allows us to extract multi-level features from a FCN [80]. The features are fed into an efficient multi-layer network, which predicts a low-dimensional affine transformation parameter vector. We then apply the affine transformation to the initial object mask to produce a refined segment candidate. Figure 5.2 illustrates the overall model structure of our approach. We now describe each module of our system in detail.

Method(Dataset)	mean PGIoU	mean RGIoU	Gain
SharpMask(Cityscapes)	0.685	0.816	19.12%
DeepMask(Cityscapes)	0.677	0.819	20.97%
MCG(Cityscapes)	0.603	0.694	15.08%
SharpMask(PASCAL VOC)	0.688	0.803	16.72%
DeepMask(PASCAL VOC)	0.671	0.803	19.67%
MCG(PASCAL VOC)	0.628	0.721	14.83%

Table 5.1: The IoU scores before and after applying the oracle affine transformation to the initial segment proposals and their relative gains. The 'mean PGIoU' denotes the average IoU score of the original proposals, while the 'mean RGIoU' is the average IoU score of the warped proposals.

5.2.1 Refinement by Affine Transformation

Our refinement method starts from an initial set of object segments generated by any existing proposal method. In order to evaluate the generality of our refinement procedure, we consider three segment proposal methods to cover different types of proposal mechanism in this work: 1) MCG [5], which is a state-of-the-art method based on hierarchical over-segmentation and ranking; 2) DeepMask [11], which is an end-to-end deep network method for segment generation; 3) SharpMask [12], one of the state-of-the-art method with its own refinement step.

We note that the initial segment candidates have a large variation in their deviations from the groundtruth object segments due to inaccurate pixel groupings. In general, it requires a rich family of nonrigid transformations to warp these initial segment masks onto the groundtruth masks. However, it is challenging to predicting such nonrigid transforms due to its complexity in model design and training procedure. In this work, we instead consider a simpler family of spatial transformations for warping the input segment masks. Specifically, we adopt the 2D affine transformation for refining the segments, which has only six degrees of freedom. Such a constrained transformation space enables us to design an efficient network to predict the required transformation parameters.

To validate the sufficiency of the affine transformation, we first compute an oracle affine transformation for each input segment mask whose Intersection-over-Union (IoU) with the ground truth is larger than 0.5, and measure the improvements on the quality of segment proposals. We use the off-the-shelf nonrigid registration toolbox [94] to compute the oracle affine transformation between an input and its nearest groundtruth mask. Table 5.1 shows the average IoU values before and after applying the oracle affine transformations, as well as its overall gains in percentage, on two public datasets. We can see that, while not perfect, the affine transformations are capable of achieving significant improvements over SharpMask, DeepMask and MCG, which shows their effectiveness for the refinement.

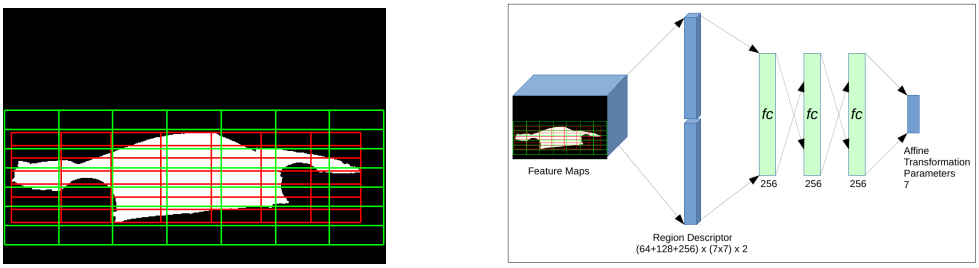


Figure 5.3: **Left**:The design of our mask feature pooling scheme for the region around an segment mask. We extract two types of features for a segment, denoted by the red and green grids respectively. See text for details. **Right**: The architecture of our regression network, which has four fully-connected layers and outputs 7 affine transformation parameters.

5.2.2 Affine Transformation Regression Network

Given an input image and an initial segment mask, we formulate the refinement as a regression problem, in which we use the image and input mask cues to predict the required affine transformation. To this end, we design a deep regression network that consists of two main components: a mask feature pooling module and a transformation regression module. We now introduce the details of these two modules as follows.

5.2.2.1 Mask Feature Pooling

Our mask feature pooling module is built on top of the FCN [80]. For an input image, we first feed it into an FCN to generate multiple convolutional feature maps for the entire image. Specifically, we adopt the FCN-8s model [80], which produces feature maps from $pool_1$ to $pool_5$ with different spatial resolutions. We take the convolutional feature maps from $pool_1$, $pool_2$ and $pool_3$ for extracting our mask features, as they encode the low- and mid-level image cues and capture the geometric information required for estimating spatial transformation[1].

We design a mask feature pooling module for each input segment candidate as in most detection networks [8]. However, as our initial segments are mostly misaligned with the groundtruth object regions, we propose a dual pooling strategy to capture both the mask information and the spatial context cue of the initial segment. Specifically, we conduct the mask feature pooling with two different receptive fields and form the segment descriptors by concatenating the two types of pooled feature representations.

The first mask feature pooling aims to capture the shape of the segment mask and the convolutional features in the segment. To achieve this, we form a tight bounding box enclosing the mask and divide it evenly into $nH \times nW = 7 \times 7$ cells (as illustrated

[1]We also investigated other settings that add $pool_4$ and $pool_5$ feature maps, but did not obtain noticeable improvements.

by the red grid in Figure 5.3 (Left)). In each cell, we adopt the convolutional feature masking [9] to compute its pooled features. Specifically, we map each cell in the image domain (where the binary mask is defined) onto each layer of feature maps, *e.g.* the $pool_1$ feature maps, according to the receptive field geometry [110]. For each mapped cell, we conduct the max-pooling in the partial mask inside the cell. If no mask overlaps with the cell, the pooling output will be $\mathbf{0}$. For $pool_k$ $(k = 1, 2, 3)$ feature maps with n_k layers, we then obtain a feature vector with $n_k \times nH \times nW$ elements after pooling, and the first pooled feature representation is formed by concatenating such feature vectors from all three types of convolutional maps.

The second mask feature pooling captures more contextual information around the initial segment. As many masks only partially cover a groundtruth object region, we consider using a larger receptive field to pool the features so that it can provide more global information for the regression network to predict the affine transformations. Concretely, for each segment, we expand the previous tight bounding box by increasing its height and width by 1.5 times. We then pool the feature representation of the larger bounding box in a similar manner to the first mask feature pooling (as illustrated by the green grid in Figure 5.3 (Left)). However, we do not use mask information here and only conduct standard max-pooling within each cell.

5.2.2.2 Regression Network Architecture

The transformation regression module takes the segment descriptor as its input and predict the affine transformation to warp the input segment mask. Instead of generating the affine transformation matrix directly, we represent the transformation by seven parameters corresponding to translation in x,y directions, rotation, scaling and shearing in x,y directions, denoted as $(t_x, t_y, r, s_x, s_y, h_x$ and $h_y)$, respectively. Formally, the 2D affine transformation T (in homogeneous coordinates) is defined as follows,

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos(r) & sin(r) & 0 \\ -sin(r) & cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

We found this parametrization leads to a better performance in practice. Our regression network consists of three fully-connected (fc) layers followed by a linear layer to output seven parameters for the predicted affine transformation. Each fully-connected layer has 256 neurons and uses ReLU as their activation functions. We also add batch normalization [112] to each layer and a dropout layer to each of the first two layers. Figure 5.3 (Right) illustrates the architecture of our network. We use the MatConvNet [110] toolbox to implement our network in this work.

5.2.3 Network Training

While our full network can be trained in an end-to-end manner, we take a two-stage training strategy due to high memory requirement in the mask feature pooling module. In particular, we first pre-train the FCN-8s model using semantic segmentation datasets (see Section 5.3 for details), which is used to compute the convolutional feature maps. In the second stage, we train the transformation regression network that maps the segment descriptors computed from the mask feature pooling module to the affine transformation parameters.

Training Data for Regression Network. The dataset for training the regression network is built as follows. From the initial object candidate set, we first select the object segments whose IoU with its corresponding groundtruth mask is greater than 0.5. The oracle affine transformations are then estimated using the nonrigid registration toolbox [94] and used as our ground truth for training the regression network. More concretely, we use a larger bounding box of the initial segment to crop a region of interest, and estimate the required warping from the initial mask to the corresponding groundtruth mask in that region. Interestingly, we also find that adding initial candidates with lower IoU scores does not improve the network performance.

Details of Training Procedure. Given the pairs of segment descriptor and affine transformation parameters, we train the transformation regression network to minimize the L_1 *loss* of the training set, which is more robust than the L_2 *loss*. We use stochastic gradient descent with a batch size of 1,024 examples, momentum of 0.9, weight decay of 0.0005 and train the network for 10 epochs. The learning rate we use for each epoch gradually decreases from 10^{-1} to 10^{-4} evenly in the log space.

5.3 Experiments

In this section, we evaluate our object segment proposal refinement method on two publicly available datasets: the Cityscapes dataset [16] and the PASCAL VOC dataset [17, 97]. Both datasets provide instance-level annotations for semantic segmentation.

5.3.1 Dataset

Cityscapes [16] For the Cityscapes dataset, we follow the exactly same setup used in Section 4.3 of Chapter 4.

PASCAL VOC [17, 97] The PASCAL VOC dataset currently contains annotations from 11,355 images taken from the PASCAL VOC 2012 dataset. For each image, it provides both category-level and instance-level segmentations for the 20 object categories in the VOC 2012 challenge. Here we use the split from [97]. In total, it consists of 8,498 training images and 2,857 validation images. We randomly select 1,000 images from the training set as our validation set and use the 2,857 original

validation images as our test set. We compute the convolutional feature maps using an FCN-8s pre-trained on this dataset.

5.3.2 Evaluation Metrics and Protocols

We employ three sets of metrics to evaluate the performance of our proposal refinement method: 1) the recall vs. number of proposals at three different IoU thresholds, including $IoU = 0.5, 0.6$ and 0.7; 2) the average recall (AR) vs. number of proposals and 3) the recall vs. IoU from 0.5 to 1 with 1,000 segment proposals.

As our goal is to refine object segment proposals, we select three state-of-the-art segment proposal generation methods to produce the initial set of segmentation proposals, which include SharpMask [12], DeepMask [11] and MCG [5]. They are also considered as the baselines for our comparison. We apply the pre-trained MCG, DeepMask and SharpMask models provided by the authors to generate their results on the two datasets.

In order to test the efficacy of our method, we learn three affine transformation regression networks for SharpMask, DeepMask and MCG respectively and apply them to the corresponding methods. Moreover, we verify the generality of our learned regression networks by applying the learned network for SharpMask to MCG proposals and the learned network for MCG to SharpMask proposals.

5.3.3 Results

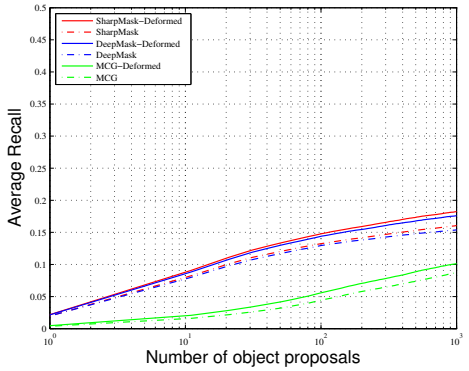
5.3.3.1 Cityscapes

In Figure 5.4(a), we first report the AR vs. number of proposals and comparisons to the baselines on the Cityscapes dataset. It shows that our approach consistently improves the quality of initial segment proposals generated by the three top-performing methods. We also achieve sizeable performance gains over these baselines. In particular, with 1,000 proposals, our method boosts the AR of SharpMask, DeepMask and MCG from 0.160, 0.154 and 0.088 to 0.182, 0.176 and 0.101 respectively and the corresponding performance gains are 13.75%, 14.29% and 14.77%.

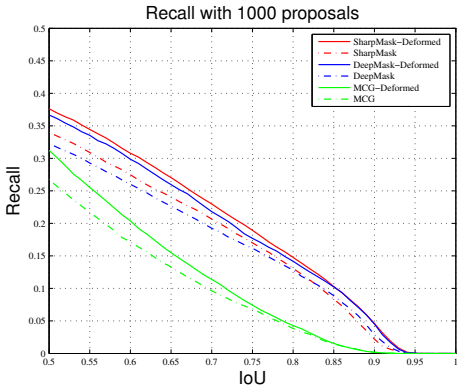
We also report the recall across different IoU thresholds with 1,000 proposals in Figure 5.4(b), which evidences that our method is capable of refining the object segmentation proposals with different qualities while maintaining the quality of segments with high IoU scores.

In Figure 5.4(c), we compare the performances (AR vs. number of proposals) of our networks when applying them to the proposals from the original initial method and a different one. We can see that the AR (0.174 for SharpMask and 0.097 for MCG) obtained by applying the learned network to the other initial proposals are just slightly lower than the original ones (0.182 and 0.101), which demonstrates the generality of our learned network.

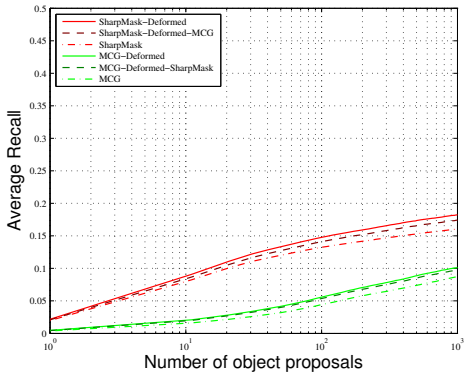
The remaining plots in Figure 5.4 describe the recalls of baselines and our method when varying the number of object proposals under different IoU thresholds. Again, they show that our approach can consistently enhance the quality of the initial object



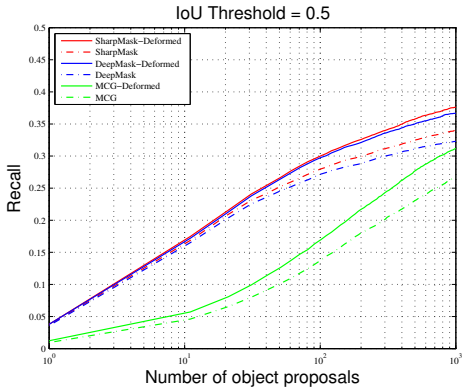
(a) AR vs. Number of proposals



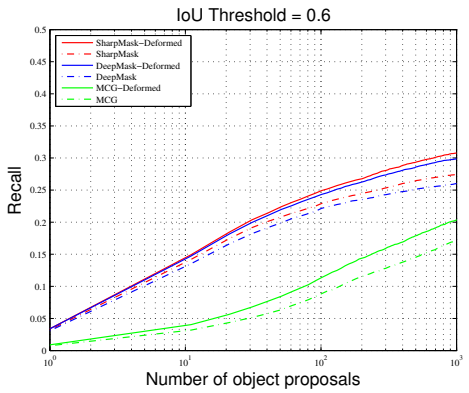
(b) Recall vs. different IoU thresholds (1,000 proposals)



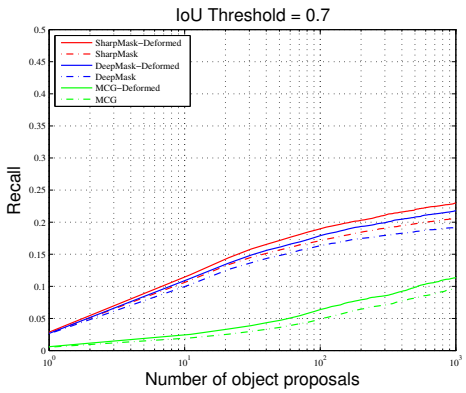
(c) AR vs. Number of proposals



(d) Recall vs. number of proposals



(e) Recall vs. number of proposals



(f) Recall vs. number of proposals

Figure 5.4: Results on **Cityscapes:** (a) and (c): average recall vs. number of proposals; (b): recall vs. different IoU thresholds for 1,000 proposals; (d), (e) and (f): Recall vs. number of proposals under different IoU thresholds (0.5, 0.6 and 0.7 respectively).

Method	AR@10	AR@100	AR@1000	AUC
SharpMask-Deformed	0.091	0.148	0.182	0.166
SharpMask	0.082	0.133	0.160	0.147
DeepMask-Deformed	0.088	0.144	0.176	0.161
DeepMask	0.080	0.130	0.154	0.143
MCG-Deformed	0.021	0.056	0.101	0.082
MCG	0.016	0.045	0.087	0.069

Table 5.2: Quantitative results on **Cityscapes**: AR at different number of proposals (10, 100 and 1,000) and AUC (AR averaged across all proposal counts).

proposals across different IoU thresholds and with different number of proposals. For example, when the IoU threshold being 0.5 (see Figure 5.4(d)), the recall improvements for SharpMask, DeepMask and MCG are 10.59% (from 0.340 to 0.376), 13.62% (from 0.323 to 0.367) and 14.93% (from 0.268 to 0.308) respectively.

More detailed quantitative results for the Cityscapes dataset are shown in Table 5.2, where we report the AR at three settings with different selected numbers of proposals, and the averaged AR across all proposal numbers (AUC). In addition, we show some qualitative examples of the mask refinement on the Cityscapes dataset in Figure 5.6. We note that our method is able to warp the initial segment masks towards the groundtruth objects through various transformations, including translation, expansion and shrinkage.

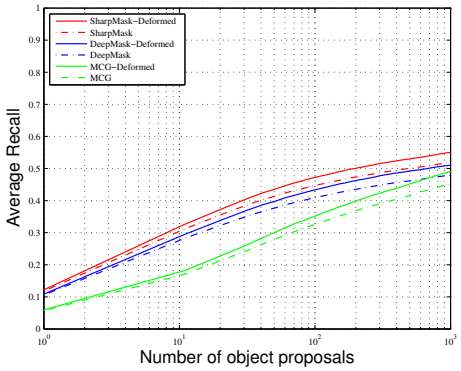
5.3.3.2 PASCAL VOC

We report the AR vs. the number of object proposals in Figure 5.5(a), which shows that our approach can improve the AR metric for three baseline methods on the PSACAL VOC dataset as well. Specifically, for the setting of 1,000 proposals, our method increases the AR of SharpMask, DeepMask and MCG by 6.17% (from 0.519 to 0.551), 6.68% (from 0.479 to 0.511) and 8.39% (from 0.453 to 0.491) respectively. We note that the quantitative improvements on the PASCAL VOC are less than those on the Cityscapes. One possible reason is that the performance of these three methods on the PASCAL VOC is better than theirs on the Cityscapes, leading to a narrower margin for improvement.

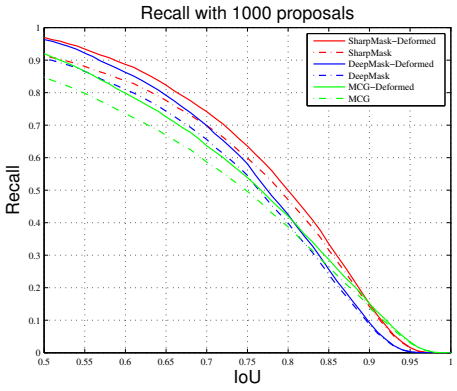
Figure 5.5(b) shows the recall changes across different IoU thresholds with 1,000 proposals. Again, we can see that the improvement for the initial object proposals is evident.

In Figure 5.5(c), we compare the original results with the ones obtained by applying the learned network to a different initial proposal method in terms of AR vs. number of proposals. The results clearly show the generality of our networks w.r.t. the initial proposal set.

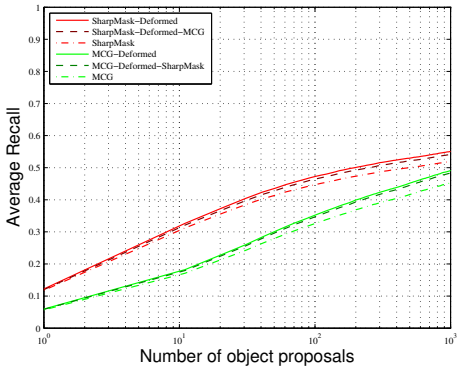
Similarly, the remaining plots in Figure 5.5 show the recall improvement under different IoU thresholds when varying the number of proposals. It demonstrates



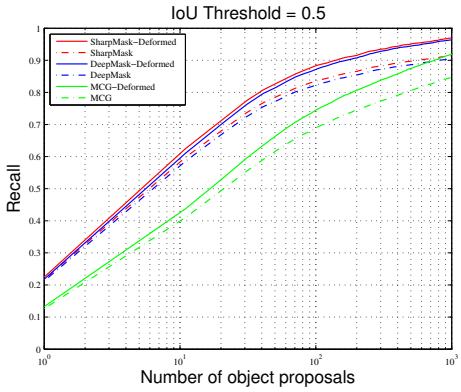
(a) AR vs. Number of proposals



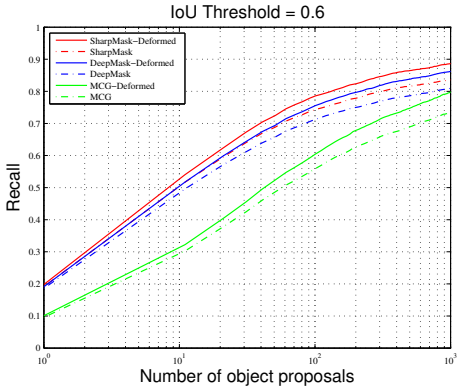
(b) Recall vs. different IoU thresholds (1,000 proposals)



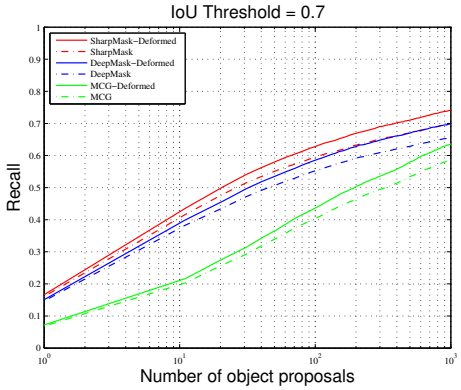
(c) AR vs. Number of proposals



(d) Recall vs. number of proposals



(e) Recall vs. number of proposals



(f) Recall vs. number of proposals

Figure 5.5: Results on **PASCAL VOC:** (a) and (c): average recall vs. number of proposals; (b): recall vs. different IoU thresholds for 1,000 proposals; (d),(e) and (f): Recall vs. number of proposals under different IoU thresholds (0.5, 0.6 and 0.7 respectively).

Method	AR@10	AR@100	AR@1000	AUC
SharpMask-Deformed	0.321	0.473	0.551	0.514
SharpMask	0.307	0.447	0.519	0.486
DeepMask-Deformed	0.292	0.434	0.511	0.476
DeepMask	0.281	0.409	0.479	0.447
MCG-Deformed	0.182	0.353	0.491	0.430
MCG	0.170	0.327	0.453	0.396

Table 5.3: Quantitative results on **PASCAL VOC:** AR at different number of proposals (10, 100 and 1,000) and AUC (AR averaged across all proposal counts).

IoU Interval	[0.3, 0.5)	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)
mean PGIoU	0.386	0.548	0.648	0.75
mean RGIoU	0.431	0.599	0.698	0.784
Gain	11.63%	9.42%	7.84%	4.57%
mean PGIoU	0.388	0.549	0.649	0.748
mean RGIoU	0.421	0.579	0.669	0.763
Gain	8.56%	5.6%	3.18%	1.95%

Table 5.4: Statistics for the improvements in the quality of DeepMask proposals with different initial IoU scores on **Cityscapes** (Top) and **PASCAL VOC** (Bottom).

again that our approach can consistently improve the quality of the original object proposals across the range of all different settings.

We report the detailed quantitative results for the PASCAL VOC in Table 5.3, which describes the AR at three settings with selected numbers of proposals and the averaged AR across all proposal numbers (AUC). Finally, some qualitative examples of the mask refinement on the PASCAL VOC dataset are shown in Figure 5.7 and Figure 5.8. Again, we can see our method achieves better region alignment for a variety of scenarios.

5.3.4 Ablation Study

To gain more insight into our approach, we conduct an ablation study by computing the improvements in the quality of DeepMask proposals with different initial IoU scores on two datasets. We first divide the initial proposals set into 4 groups, which correspond to the IoU intervals of [0.3, 0.5), [0.5, 0.6), [0.6, 0.7) and [0.7, 0.8). We then compute the mean IoU improvements for each group after warping the initial proposals through our method, which are shown in Table 5.4. The results show that our method is more effective on correcting large errors than obtaining fine-grained details, which is most likely due to the coarse-level warping generated by the affine transformations.

5.4 Conclusion

In this chapter, we propose a novel method for refining object segment proposals, which can generate object segment candidates with better quality for instance-level semantic segmentation. The main contribution of our work is to formulate the refinement as a regression problem that estimates 2D affine transformations to warp the initial segment masks towards groundtruth objects. We design and train a deep network to predict the affine transformation parameters based on a new mask pooling strategy defined on hypercolumn features. Extensive experimental evaluations on two challenging datasets, the Cityscapes and the PASCAL VOC, demonstrate that our approach can consistently achieve improvements on the IoU quality of the object segment proposals over three state-of-the-art methods.

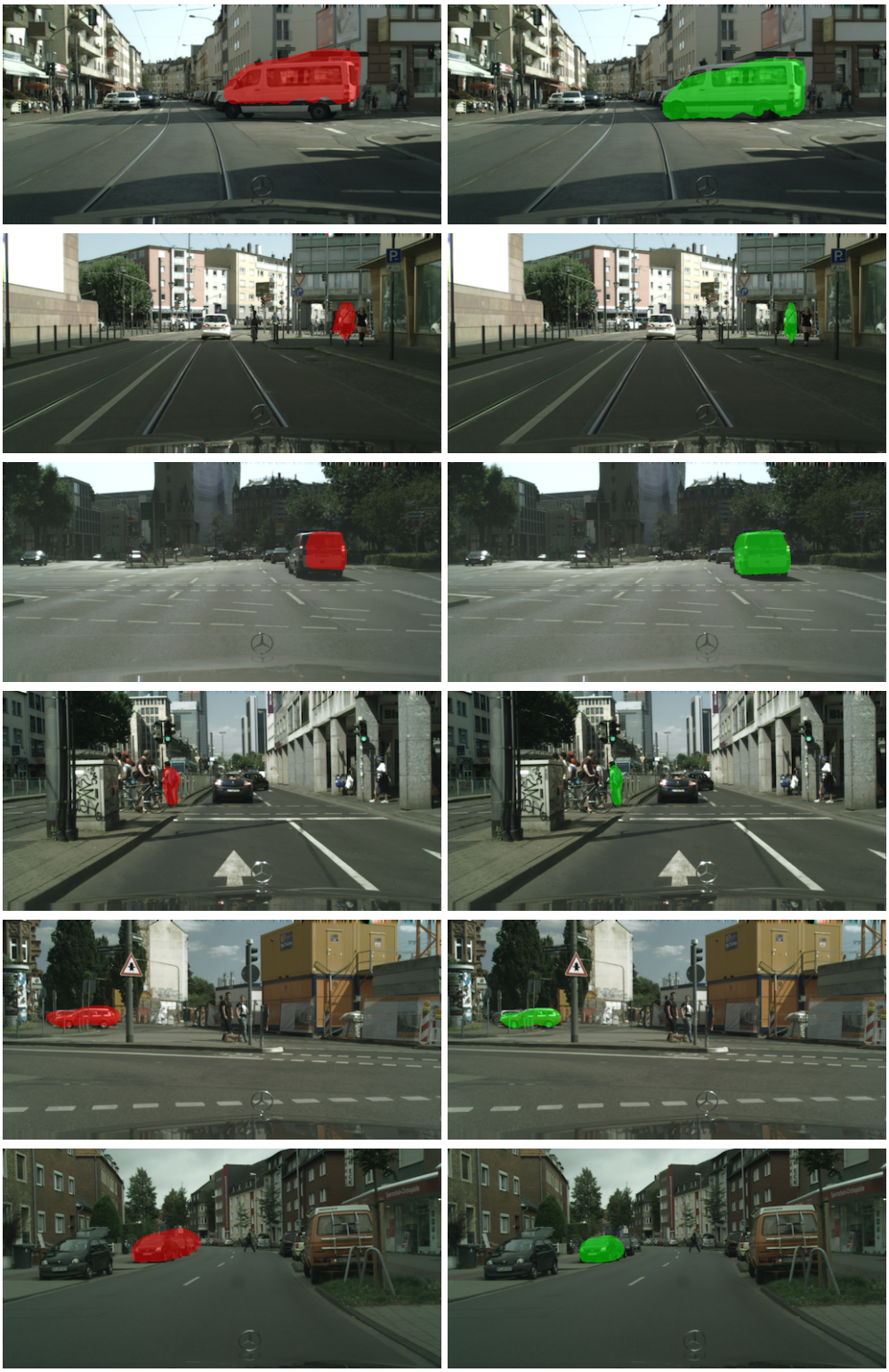


Figure 5.6: Qualitative results on **Cityscapes**. Red: original proposal's mask. Green: transformed mask.

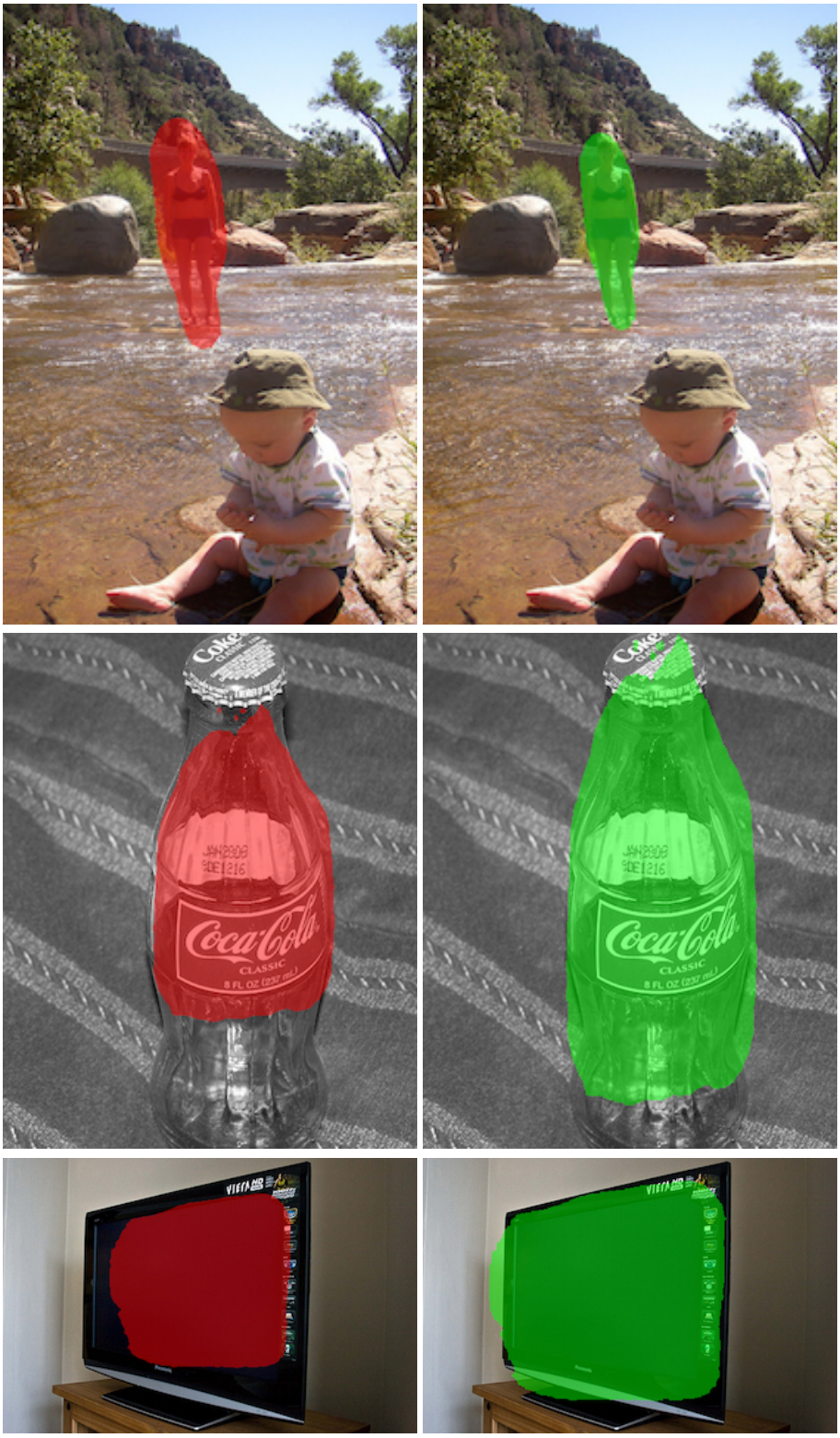


Figure 5.7: Qualitative results on **PASCAl VOC**. Red: original proposal's mask. Green: transformed mask.

Figure 5.8: Qualitative results on **PASCAl VOC**. Red: original proposal's mask. Green: transformed mask.

Chapter 6

Deep Free-Form Deformation Network for Object-Mask Registration

6.1 Introduction

In the previous chapter, we propose an affine transformation regression network to refine an initial set of segment proposals. While the affine transformation can effectively represent globe transformations at a coarse level, it lacks the capacity of describing more complex non-rigid deformations from the proposals to the corresponding ground truths. In this chapter, we employ a more flexible deformation representation, the free-form deformation model, to address such limitation, and consider the task from a novel object-mask registration perspective.

Aligning a shape mask to object instances is a commonly used strategy in segmenting objects from background or inferring shape deformation of individual objects, and has wide applications in semantic instance segmentation [59], object proposal generation [58] and visual object tracking [113], *etc*. While it can be viewed as a special case of image registration problem [114], such object-mask alignment task is more challenging as the mask lacks internal structure for finding the dense correspondence between the target object and itself.

Most existing approaches address this problem by formulating it as an object segmentation task, in which the shape mask is used as an initialization, such as contour matching [55], or an instance shape prior for binary object segmentation [56, 60]. However, the resulting segmentation task is usually equally challenging, and does not provide shape alignment between mask and object.

An alternative, and sometimes more natural approach to the object-mask alignment problem is to predict a 2D spatial transformation that registers mask onto the target object, as shown in Figure 6.1. Such a transformation-based strategy has several advantages in practice. First, the problem of predicting 2D transforms is typically simpler due to the fact that the common transformation families, such as affine or TPS [115], have fewer degrees of freedom and thus the output of prediction lies in a lower dimensional space. Second, for slightly mis-aligned mask and object,

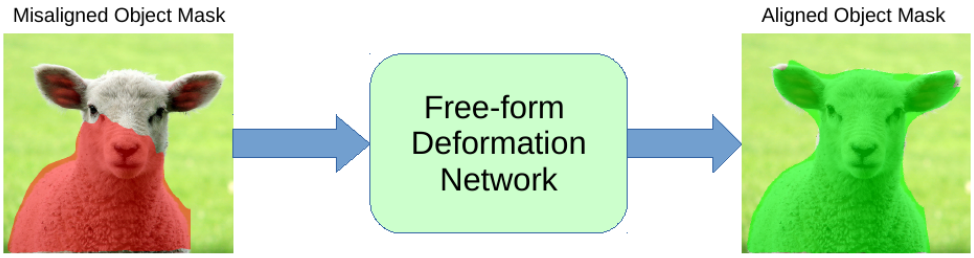


Figure 6.1: An illustration of the object-mask alignment problem and the transformation implemented by the deep free-form deformation network.

transforming binary masks is more efficient than recomputing the segmentation or doing image registration. Finally, the predicted transformation allows us to infer the detailed shape deformation of an instance relative to its canonical shape mask.

In this chapter, we propose a deep learning approach to address the object-mask alignment problem and apply it to the segment proposal refinement task. Given an input image containing the target object and an initial mask, our approach learns a non-rigid 2D transform that warps the mask onto the target object. To achieve this, we design a novel spatial transformer network that predicts a free-form deformation (FFD) [94] transform and applies the non-rigid transform to the input mask to generate a better alignment between the mask and object.

Specifically, we build a deep convolutional neural network consisting of two modules. The first module computes the convolutional feature maps from the input image, and extracts a feature representation of the image region covered by the mask. To encode the shape information of the initial mask, and the image cues around object, we develop a multi-level dual mask feature pooling method to capture the misalignment between the mask and object. Based on the multi-level features, the second network module predicts a FFD transform parameterized by the offsets of predefined control points through regression. It then applies the B-spline based FFD transform to the initial mask based on a grid generator and a bilinear sampler, which produces the final warped object mask.As these two network modules are differentiable, we train the entire deformation network in an end-to-end fashion using L_2 matching loss.

We evaluate our FFD network on a challenging object-mask alignment task, in which we aim to refine a set of object segment proposals generated from state-of-the-art methods. Our results show that we achieve improvements in Average Recall on the Cityscapes, the PASCAL VOC and the MSCOCO datasets for different initial proposal methods, which validates the efficacy of our deep FFD network.

The main contributions of our work are three folds: First, we design a novel FFD spatial transformer network to address the object-mask alignment problem. Second, our FFD deformation network is capable of capturing complex non-rigid deformations, and is fully differential that can be trained in an end-to-end manner. Finally, our method achieves consistent sizeable improvements over several stat-of-the-art approaches on challenging benchmarks.

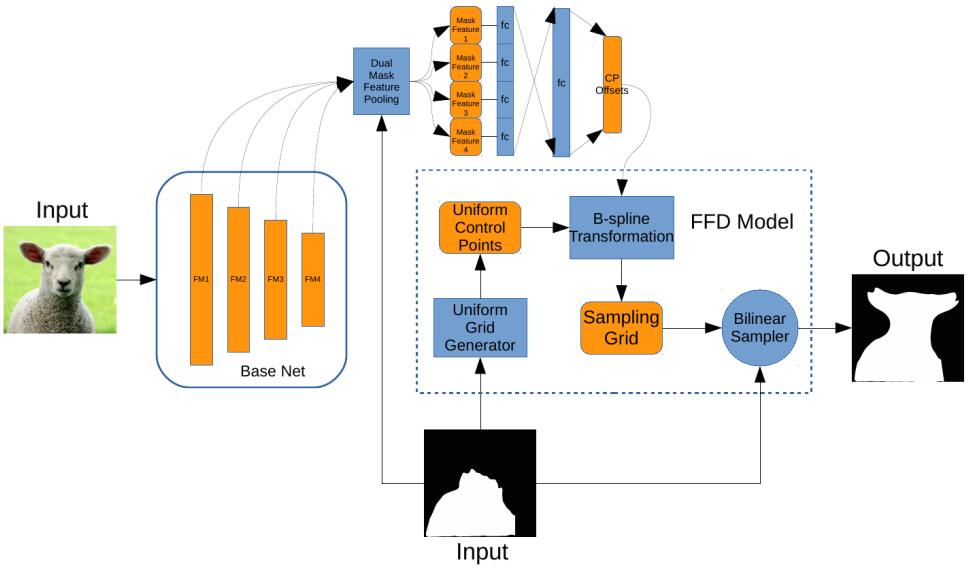


Figure 6.2: An overview of our deep FFD network for object-mask alignment. The entire network consists of two modules: the first computes the convolutional feature maps and extracts mask features using dual mask pooling, while the second predicts the FFD transform and warps the input mask onto the target object.

6.2 Deep Free-form Deformation Network

We aim to generate an object segmentation by aligning an initial mask to its target object in an input image. To this end, we take the transformation-based strategy that learns a 2D spatial transformer to warp the initial mask to the target object. In this section, we introduce a deep convolutional neural network that first predicts a non-rigid transformation and then applies the transform to the initial mask to produce the aligned object mask. Our network is fully differentiable and can be trained in an end-to-end fashion.

More specifically, our network consists of two modules: the first computes convolutional feature maps and extracts multi-level features to capture the misalignment between the mask and object, while the second module predicts the non-rigid transformation and warps the initial mask. Figure 6.2 illustrates the overview of our network structure. We now describe each module of our system in detail.

6.2.1 Convolutional Features and Mask Pooling

Our first network module uses a base convolutional neural network (CNN) to compute the convolutional feature maps of the input image. To capture the misalignment between the initial mask and its target object, we introduce a dual mask feature pooling scheme to extract multi-level features from the feature maps. In particular, this scheme enables us to capture the mask shape information and the spatial context cue

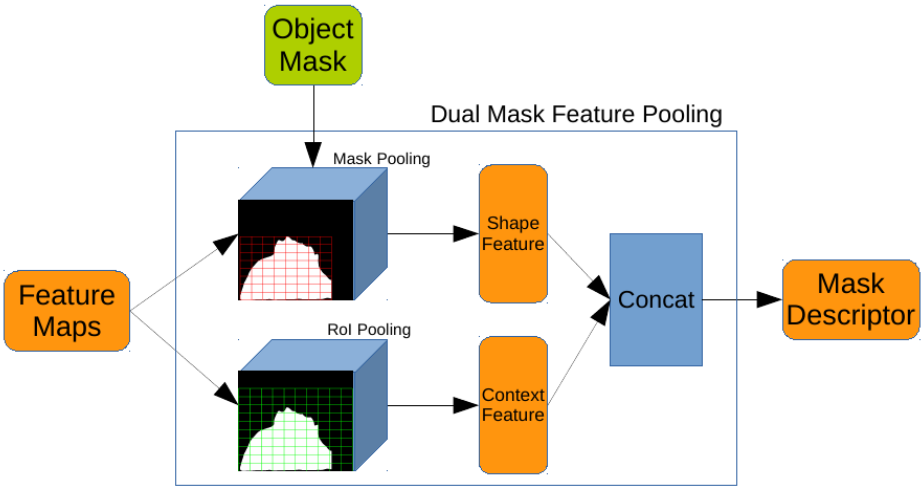


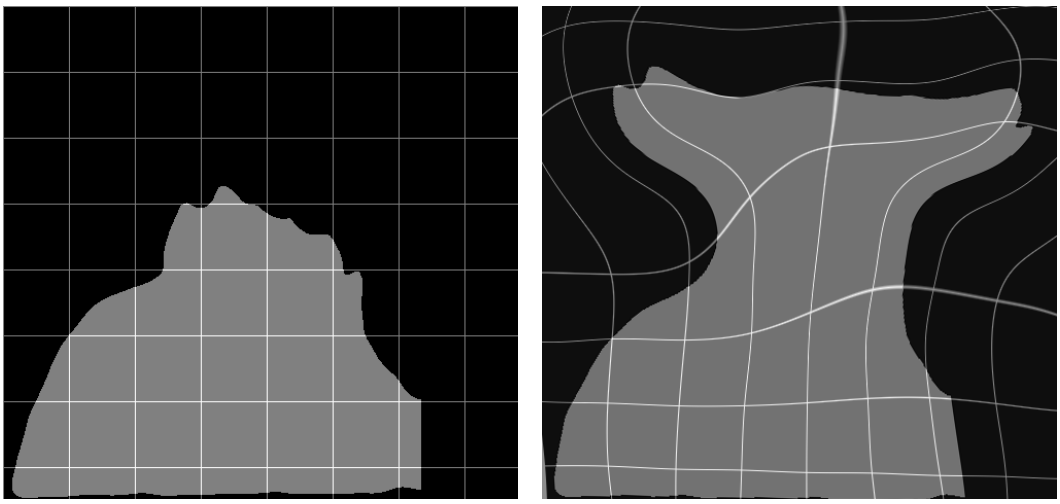
Figure 6.3: The dual mask feature pooling pipeline in our FFD network. Here only a single level of convolutional maps is shown. Note that we use much finer grid partition than the standard RoI pooling.

around the object region that can guide the network to predict the spatial warping.

Our pooling layer takes as input a set of convolutional feature maps and an object mask, and generates an object-mask descriptor. Its design is inspired by the standard RoI pooling [8] and the convolutional feature masking [9] methods. Specifically, we form a tight bounding box enclosing the mask as well as a larger box by expanding the tight box in its height and width directions by 1.6 times. We first do weighted RoI pooling in the tight box, where the output of the standard RoI pooling in each cell is weighted by the overlap ratio between the cell and the mask. This generates the first type of mask features, encoding the shape and the convolutional features covered by the mask. We then perform the standard RoI pooling in the larger bounding box. This second type of features captures the spatial context cue of the mask and the target object. The final object-mask descriptor is formed by concatenating the two types of pooled mask features. Note that different from the RoI pooling in object detection [8], we compute the mask feature pooling on all convolution feature maps generated by the base network (as shown in Figure 6.2), which allows us to capture both local and global cues for predicting the transformation. Figure 6.3 illustrates the dual mask feature pooling process for a single level of feature maps.

6.2.2 Free-form Deformation Transformer

Given the object-mask descriptor, our second network module predicts a 2D spatial transform to warp the initial mask onto the target object. As the mask can have arbitrary shapes, we adopt a rich family of spatial transforms, which is capable of representing any non-rigid warping in image, referred to as free-form deformation



(a) Uniformly spaced control points (b) Deformation by shifting the control points

Figure 6.4: Illustration of FFD defined on a binary mask. Left is the original mask with uniformly spaced control points; Right is the deformed mask with displaced control points.

(FFD) [94].

The FFD defines a family of non-rigid spatial transformations based on a mesh of control points. By shifting the control points and interpolating the dense deformation based on B-splines [93], it provides a flexible tool to describe the non-rigid transformation between the mask and object. Figure 6.4 shows an example of the deformation process.

Formally, let Φ be a 2-D mesh of control points and $T : (x, y) \mapsto (x', y')$ be a pointwise transformation of any location (x, y) in target image F to the location (x', y') in the source image R. Given a mesh of control points $\phi_{i,j}$ with uniform spacing δ pixels, the non-rigid transformation T by B-spline functions is defined by

$$T_{(x,y)} = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(u) B_m(v) \phi_{i+l,j+m} \tag{6.1}$$

where $i = \lfloor x/\delta \rfloor - 1$, $j = \lfloor y/\delta \rfloor - 1$, $u = x/\delta - \lfloor x/\delta \rfloor$, $v = y/\delta - \lfloor y/\delta \rfloor$, and B_l represents the l-th basis function of cubic B-splines [93]:

$$B_0(u) = (1-u)^3/6, \quad B_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6, \quad B_3(u) = u^3/6$$

From Equation (6.1), we note that the B-spline based FFD is locally controlled as each control point $\phi_{i,j}$ affects only its $4\delta \times 4\delta$ neighborhood. This indicates that the FFD can describe highly local transformation, which is required for capturing the complex non-rigid deformations between the mask and object. Additionally, the

degree of non-rigid deformations can be controlled by changing the resolution of the mesh of control points Φ. A larger spacing of control points allows modelling of global and coarse deformations, while a small spacing of control points allows modelling of local and fine-grained deformations.

By shifting the locations of the control points from the uniform grid $\phi_{i,j}$ to $\phi_{i,j} + \Delta\phi_{i,j}$, the B-spline based FFD generates a non-rigid transformation as follows:

$$T_{(x,y)} = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(u) B_m(v) (\phi_{i+l,j+m} + \Delta\phi_{i+l,j+m}) \tag{6.2}$$

In this work, we parameterize the FFD by the offsets of its control points $\{\Delta\phi_{i,j}\}$, and our second network module first regresses the control point offsets from the object-mask descriptor. To achieve scale-invariance, we normalize the offsets by the size of the initial mask. Our transform regressor module consists of 3 fully connected (fc) layers and its outputs are the offset vectors of every control point.

To obtain the warped mask, we follow a similar strategy as the Spatial Transformer Network [13]. Given the predicted offsets, we compute the dense transformation according to Equation (6.2). The transform T then generates a sampling grid G, which is a set of points where the initial mask should be sampled in order to produce the warped mask. Next, a bilinear sampling layer takes the sampling grid and the initial mask as inputs and produces the final warped mask. We refer the reader to Section 2.6.2 for more details about the bilinear sampling process, especially the back propagation of the loss through the sampling mechanism.

We note that for the FFD transformer network, the gradients of loss L with respect to $\Delta\phi_{i,j}$ can be computed by:

$$\begin{aligned} \frac{\partial L}{\partial \Delta\phi_{i,j}} &= \frac{\partial L}{\partial G} \cdot \frac{\partial G}{\partial \Delta\phi_{i,j}} \\ &= \frac{\partial L}{\partial G} \cdot \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(u) B_m(v) \end{aligned} \tag{6.3}$$

where $\frac{\partial L}{\partial G}$ is the gradients of loss L with respect to the sampling grid G. This equation shows that given $\frac{\partial L}{\partial G}$, $\frac{\partial L}{\partial \Delta\phi_{i,j}}$ can be computed efficiently by convolution, with the filter weights as $B_l(u)B_m(v)$ and the stride being the spacing of control points δ. The differentiable property of the FFD transformer network allows loss gradients to flow back to the feature maps, which enables us to train the network in an end-to-end fashion.

6.2.3 Network Details and Training

Network Architecture. We choose ResNet-101 [116] pre-trained on the ImageNet dataset [82] for image classification task as our base net to learn the feature representation. We remove all the layers on top of *res4b22_bracnch2a_relu*, as the output from

these layers are not used in our system.

For the mask feature pooling, we select a 30×30 grid for computing the feature on the feature maps output from layer *conv1_relu (64 channels)* and layer *res2c_relu (256 channels)*, and a 20×20 grid for layer *res3b3_branch2a_relu (128 channels)* and layer *res4b22_bracnch2a_relu (256 channels)*. We discover that the high resolution of the pooling grid is important for training the network, as the non-rigid transformations to be learned by the network are highly complex, which need quite discriminative and fine features to represent them.

As the mask features pooled from different layers are of different spatial sizes and channel depths, we first fully connect each set of them into a low dimensional output of size 128 and then concatenate all the outputs together to form a feature vector of size 512. Next are another two fc layers for predicting the offsets of the control points. The weight sizes of these two *fc* layers are 512×512 and $512 \times 2 \times 13 \times 13$ respectively, which means the resolution of the mesh of control points is 13×13 in our experiments. All the fc layers except the last one are followed by a ReLU layer and a dropout layer.

Training Examples. To build the set of training examples, we select those segment proposals who have an IoU with the ground truth greater than 0.5 as the training samples. Specifically, for a qualified segment proposal, we crop it with a larger box whose size is $1.6\times$ to the tight box that encloses the segment in terms of height and width, so that the cropped region can cover more of the ground-truth object mask. We also use this large box to crop corresponding ground-truth mask as this region's ground truth.

Learning Details. We train the network to simply minimize the L_2 loss between the candidate's mask and the ground truth's, which we find is robust and effective. In fact, we find our FFD network can learn spatially smooth transformations, even without using spatial regularization terms. We adopt an image-centric training policy [8]. In our system, the mini-batch size is 1 and for every image we randomly sampled 128 training segments. Except the ResNet layers, the extra fc layers are initialized randomly from Gaussian distribution. We train the network for 10 epochs using a momentum of 0.9 and weight decay of 0.002. The learning rate we use for each epoch gradually decreases from 10^{-4} to 10^{-7} evenly in the log space.

6.3 Experiments

We apply our FFD network to the segment proposal refinement task in which we intend to improve a set of object segment proposals generated from state-of-the-art methods. We evaluate the performance of our approach on three public datasets: Cityscapes [16], PASCAL VOC 2012 [17, 97] and MSCOCO [1].

Method	AR@10	AR@100	AR@1000
MNC-r	0.052	0.131	0.180
MNC	0.041	0.102	0.136
SharpMask-r	**0.103**	**0.175**	**0.215**
SharpMask	0.085	0.141	0.171
DeepMask	0.082	0.138	0.164
MCG	0.016	0.046	0.091

Table 6.1: Quantitative results of segment proposal refinement on **Cityscapes**: AR at different number of proposals (10, 100 and 1,000).

6.3.1 Evaluation Metrics and Protocols

For performance evaluation, we compute the average recall (AR) [98] between IoU 0.5 and 0.95 for a fixed number of segment proposals. The AR metric describes the overall quality of object proposals and has been shown to correlate highly with the detection accuracy in [98]. Additionally, we report the recall versus IoU threshold for different number of proposals.

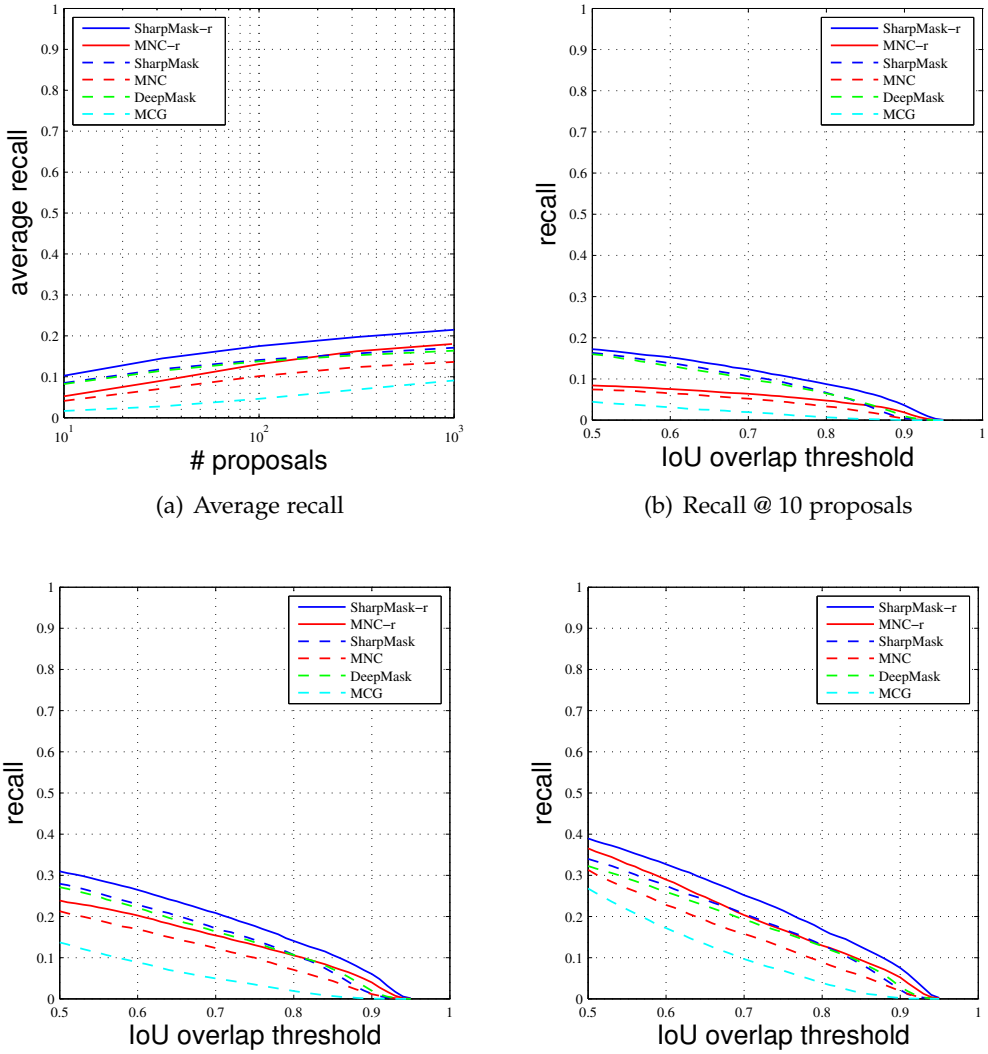
For the Cityscapes dataset, we follow the exactly same setup used in Section 4.3 of Chapter 4. For the PASCAL VOC dataset, we train our network on the training set (5,623 images) and evaluate on the validation set (5,732 images). We use the instance-level segmentation annotations from [97]. For the MSCOCO dataset, we follow the same protocol as in SharpMask [12].

To demonstrate the generality of our method, we conduct our Cityscapes and PASCAL VOC experiments with two different sets of initial object segments, which are generated from the state-of-the-art segment proposal generation methods, SharpMask [12] and MNC [3], respectively. For each type of initial segments, we train our model from scratch with a set of selected segment proposals from the initial pool. However, when training the network with SharpMask proposals on the PASCAL VOC, we find that it is difficult for the network to converge, which might be due to much fewer training segments and their sparse spatial distribution. So for that case, we fine-tune the network that has been trained for MNC proposals on the PASCAL VOC. On the MSCOCO, we only report our experiment with the SharpMask proposals.

6.3.2 Results

6.3.2.1 Cityscapes

In Fgure 6.5(a), we first report the AR performances of the refined segment proposals (**MNC-r** and **SharpMask-r**), and compare the performance of our method against the original proposal methods as well as other baselines (DeepMask [11] and MCG [5]) on the Cityscapes. We can see that our FFD network can improve the quality of the initial segment proposals by a clear margin. Specifically, with 1,000 proposals, our



(a) Average recall

(b) Recall @ 10 proposals

(c) Recall @ 100 proposals

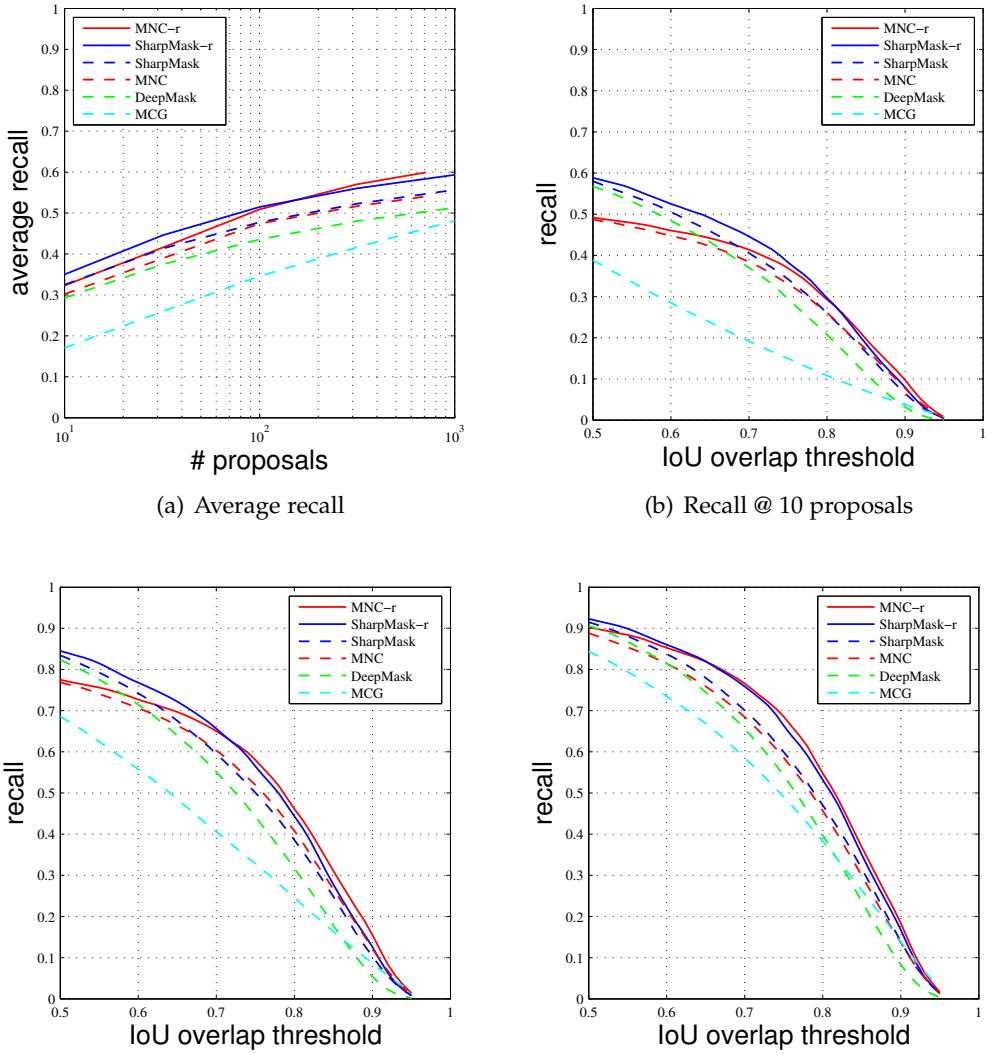
(d) Recall @ 1000 proposals

Figure 6.5: Segment proposal refinement results on **Cityscapes**: (a) AR vs. number of proposals; (b), (c) and (d) recall vs. IoU threshold with different number of proposals.

FFD network increases the AR of MNC and SharpMask from 0.136 to 0.180 (32.4% improvement) and from 0.171 to 0.215 (25.7% improvement), respectively. More detailed quantitative results are shown in Table 6.1.

Figure 6.5(b), 6.5(c) and 6.5(d) show the recall versus IoU threshold with 10, 100 and 1,000 proposals respectively. They demonstrate that our method can improve the proposals with different segmentation qualities on the Cityscapes dataset.

We further report some qualitative results in Figure 6.8 and Figure 6.9. These examples show that our FFD network is capable of predicting non-rigid deformations for both local and global warping, and produces better segmentation for the target



(a) Average recall

(b) Recall @ 10 proposals

(c) Recall @ 100 proposals

(d) Recall @ 1000 proposals

Figure 6.6: Segment proposal refinement results on **PASCAL VOC**: (a) AR vs. number of proposals; (b), (c) and (d) recall vs. IoU threshold with different number of proposals.

objects with different scales and classes.

6.3.2.2 PACAL VOC

We compare the AR performances of our method with other baselines on the PASCAL VOC in Figure 6.6(a). It can be seen that our FFD network further improves the quality of the segment proposals generated from both state-of-the-art approaches. In particular, with 1,000 proposals, our FFD network increases the AR of MNC and SharpMask by 10.52% (from 0.542 to 0.599) and 6.64% (from 0.557 to 0.594). More

Method	AR@10	AR@100	AR@1000
MNC-r	0.323	0.509	**0.599**
MNC	0.302	0.474	0.541
SharpMask-r	**0.350**	**0.515**	0.594
SharpMask	0.325	0.477	0.557
DeepMask	0.293	0.436	0.513
MCG	0.171	0.346	0.481

Table 6.2: Quantitative results of segment proposal refinement on **PASCAL VOC** : AR at different number of proposals (10, 100 and 1,000).

Method	AR@10	AR@100	AR@1000
SharpMask-r	**0.179**	**0.327**	**0.416**
SharpMask	0.160	0.298	0.387

Table 6.3: Quantitative results of segment proposal refinement on **MSCOCO** : AR at different number of proposals (10, 100 and 1,000).

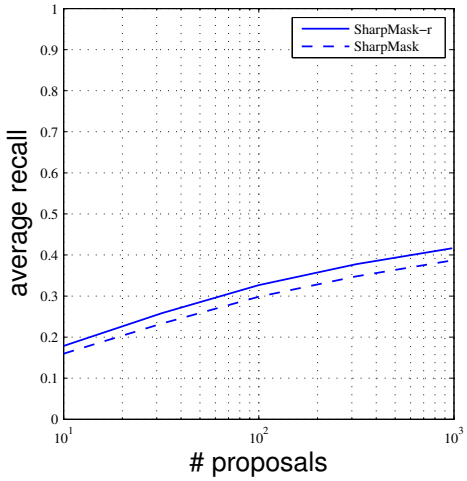
IoU Interval	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)	[0.8, 0.9)
mean PGIoU	0.548	0.648	0.749	0.849
mean RGIoU	0.665	0.737	0.796	0.861
Gain	21.35%	13.7%	6.28%	1.41%

Table 6.4: Statistics for the improvements in the quality of MNC proposals with different initial IoU scores on **PASCAL VOC**. The 'mean PGIoU' denotes the average IoU score of the original proposals, while the 'mean RGIoU' is the average IoU score of the warped proposals.

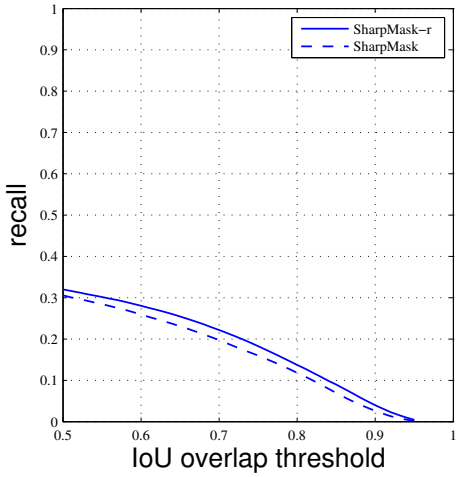
detailed quantitative results are shown in Table 6.2. This demonstrates that our approach generalizes well to other types of datasets.

Figure 6.6(b), 6.6(c) and 6.6(d) show the recall versus IoU threshold with 10, 100 and 1,000 proposals respectively. We can see that the refined proposals have better quality, as with high IoU thresholds, *e.g.* 0.7, 0.8 and 0.9, the refined proposals have much higher recall than the initial proposals.

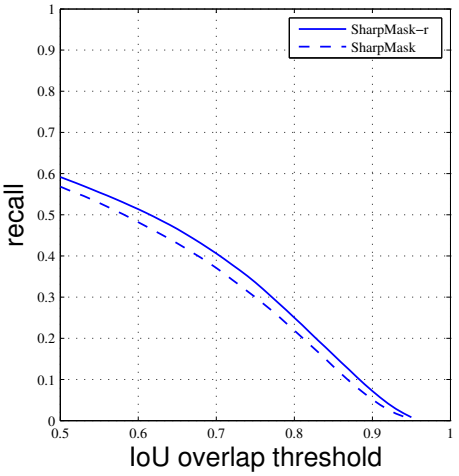
Additionally, we include selected qualitative examples in Figure 6.10 and Figure 6.11, which show that our FFD network produces a wide range of refinements on object shapes. Some of these results have a slightly better boundary alignment, while the others achieve large improvements over the initial segments.



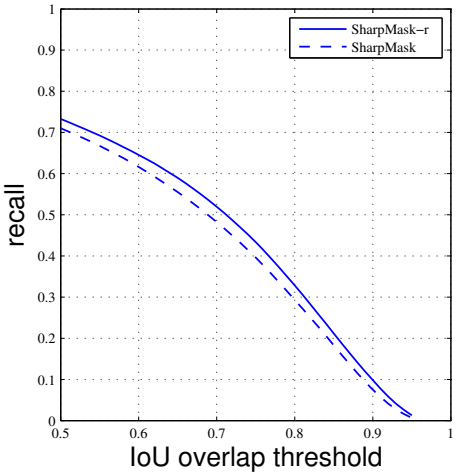
(a) Average recall



(b) Recall @ 10 proposals



(c) Recall @ 100 proposals



(d) Recall @ 1000 proposals

Figure 6.7: Segment proposal refinement results on **MSCOCO**: (a) AR vs. number of proposals; (b), (c) and (d) recall vs. IoU threshold with different number of proposals.

6.3.2.3 MSCOCO

Figure 6.7(a) demonstrates the AR improvement for SharpMask proposals on the MSCOCO, while Tabel 6.3 shows more detailed quantitative results. With 1,000 proposals, our approach improve the AR by 7.49% (from 0.387 to 0.416). Figure 6.7(b), 6.7(c) and 6.7(d) show the recall versus IoU threshold with 10, 100 and 1,000 proposals respectively. It is clear that our method can achieve consistent improvements on MSCOCO, and this demonstrates that our approach is able to scale up to a larger number of object classes. Figure 6.12 shows a few selected qualitative examples.

Method	mAP@0.50	mAP@0.75	Average mAP
SharpMask-r	**0.461**	**0.127**	**0.195**
SharpMask	0.448	0.122	0.190

Table 6.5: Fast RCNN results on **PASCAL VOC** : mAP at different IoU thresholds (0.5 and 0.75) and average mAP across IoU thresholds (0.5:0.05:0.95) with 1,000 proposals.

Method	ACCV	SharpMask	WACV	ICCV
AR@1000	0.150	0.160	0.182	**0.201**
AR@100	0.099	0.133	0.148	**0.162**

Table 6.6: Longitudinal comparison of our segment proposal methods on **Cityscapes** : AR at 1,000 and 100 proposals.

6.3.3 Ablation Study

To get more insight into our FFD network, we analyze the IoU improvements for MNC segment proposals with different IoU scores on the PASCAL VOC. We divide the initial proposal set into 4 groups, which correspond to the IoU intervals of [0.5, 0.6), [0.6, 0.7), [0.7, 0.8) and [0.8, 0.9). We then compute the mean IoU improvements for each group after aligning the initial masks to their object regions through the FFD network. The results are shown in Table 6.4, from which we can see our FFD network is more effective in modeling relatively coarse transformations than capturing fine-level local deformations. Encoding such fine-level misalignment between the mask and its groundtruth might require finer features and denser control points.

We have also tried to learn a backward transformation that warps the groundtruth mask to the proposal mask. Interestingly, we discover that the backward transformation is much easier to learn, which can be explored further in future work.

6.3.4 Object Detection

As a final validation, we evaluate how our refined proposals perform for object detection. We take the off-the-shelf Fast RCNN [8] model trained on the PASCAL VOC 2007 with SelectiveSearch [40] proposals as our detection system. To show the generalization of our approach, we do not retrain this detection network with our proposals and directly apply the pre-trained detection model to our proposals. We evaluate the refined and initial SharpMask proposals on the PASCAL VOC 2012 validation set and compare their detection performances. We take the bounding boxes tightly enclosing the segment masks as input and report the mean average precision (mAP) when varying the IoU threshold from 0.5 to 0.95 with 1,000 proposals.

Table 6.5 shows the detection results. Our refined proposals outperforms the initial proposals, improving the average mAP from 0.190 to 0.195. At different IoU thresholds, our proposals consistently perform better than the initial proposals. This shows that the improvements on the quality of proposals obtained by our method

can benefit the object detection performance.

6.3.5 Longitudinal Comparison

In this section, we conduct a comprehensive comparison on the quality of proposals generated by three of our main methods. Specifically, Table 6.6 shows the results of our ACCV work (Charpter 4), WACV work (Charpter 5) and ICCV work (Charpter 6) on the Cityscapes dataset. We can see that our ICCV work achieves the best performance. With 1,000 proposals, it improves the AR of initial SharpMask proposals by 25.7% (from 0.160 to 0.201), which is almost 2x the 13.8% improvement (from 0.160 to 0.182) obtained in our WACV work. This indicates that our FFD deformation network is much more powerful than the previous affine transformation regression network. Besides, the performances of our last two methods are obviously better than our ACCV work's. This demonstrates that our series of works have gradually advanced the performance of object segment proposal generation. Note that we compute the AR between IoU 0.5 and 1 for this comparison, to be consistent with the evaluation metrics used in previous works.

6.4 Conclusion

In this work, we address the problem of object-mask registration and aim to align a shape mask to a target object instance. To this end, we take a transformation based approach that predicts a 2D non-rigid spatial transform and warps the shape mask onto the target object. In particular, we propose a deep spatial transformer network that learns free-form deformations (FFDs) to non-rigidly warp the shape mask based on a multi-level dual mask feature pooling strategy. Our network is fully differentiable and thus can be trained in an end-to-end manner. We evaluate our FFD network on the task of refining a set of object segment proposals, and our approach achieves the state-of-the-art performance on the Cityscapes, the PASCAL VOC and the MSCOCO datasets.

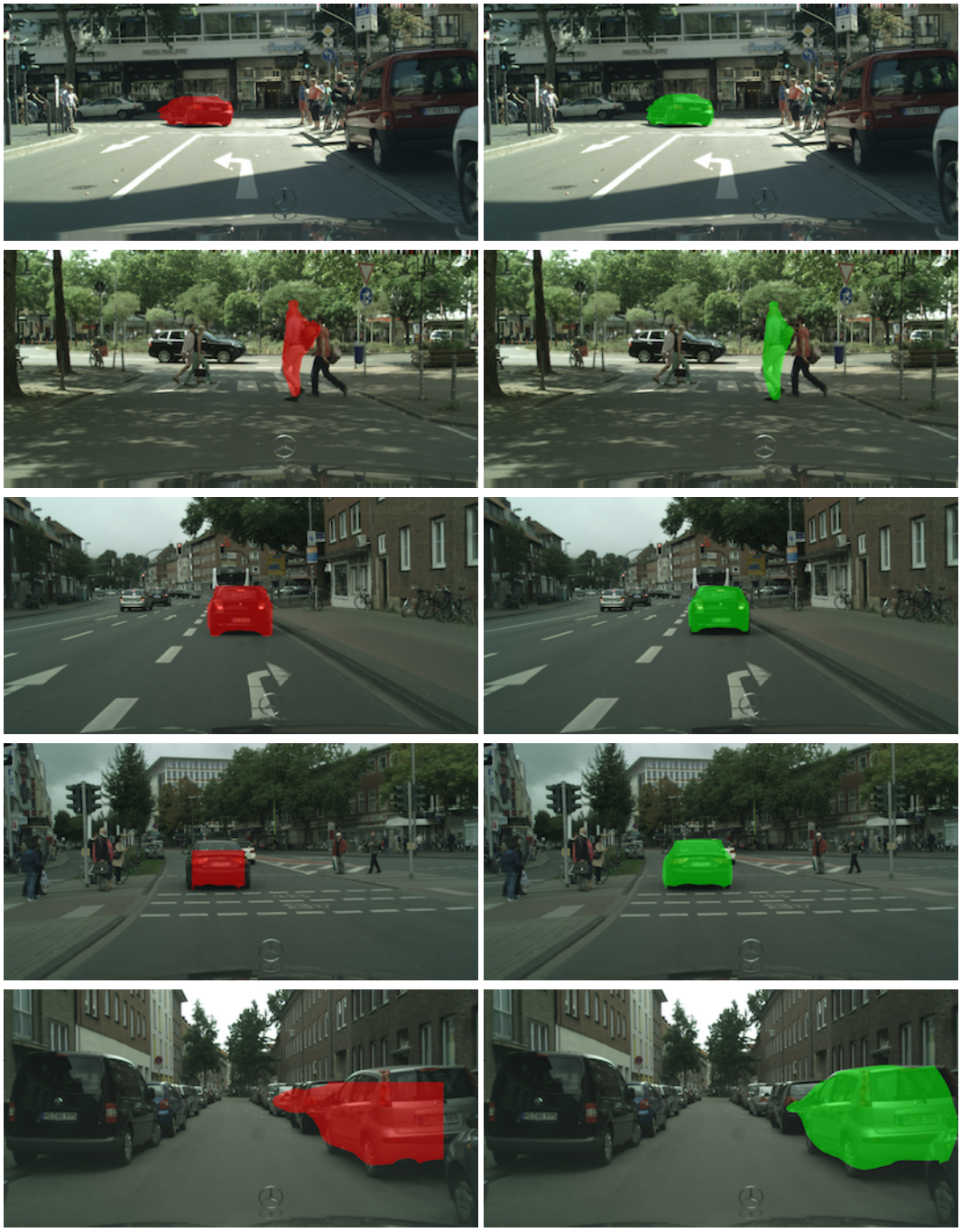


Figure 6.8: Qualitative examples for segment proposal refinement on **Cityscapes**. Red: original object mask. Green: aligned mask.

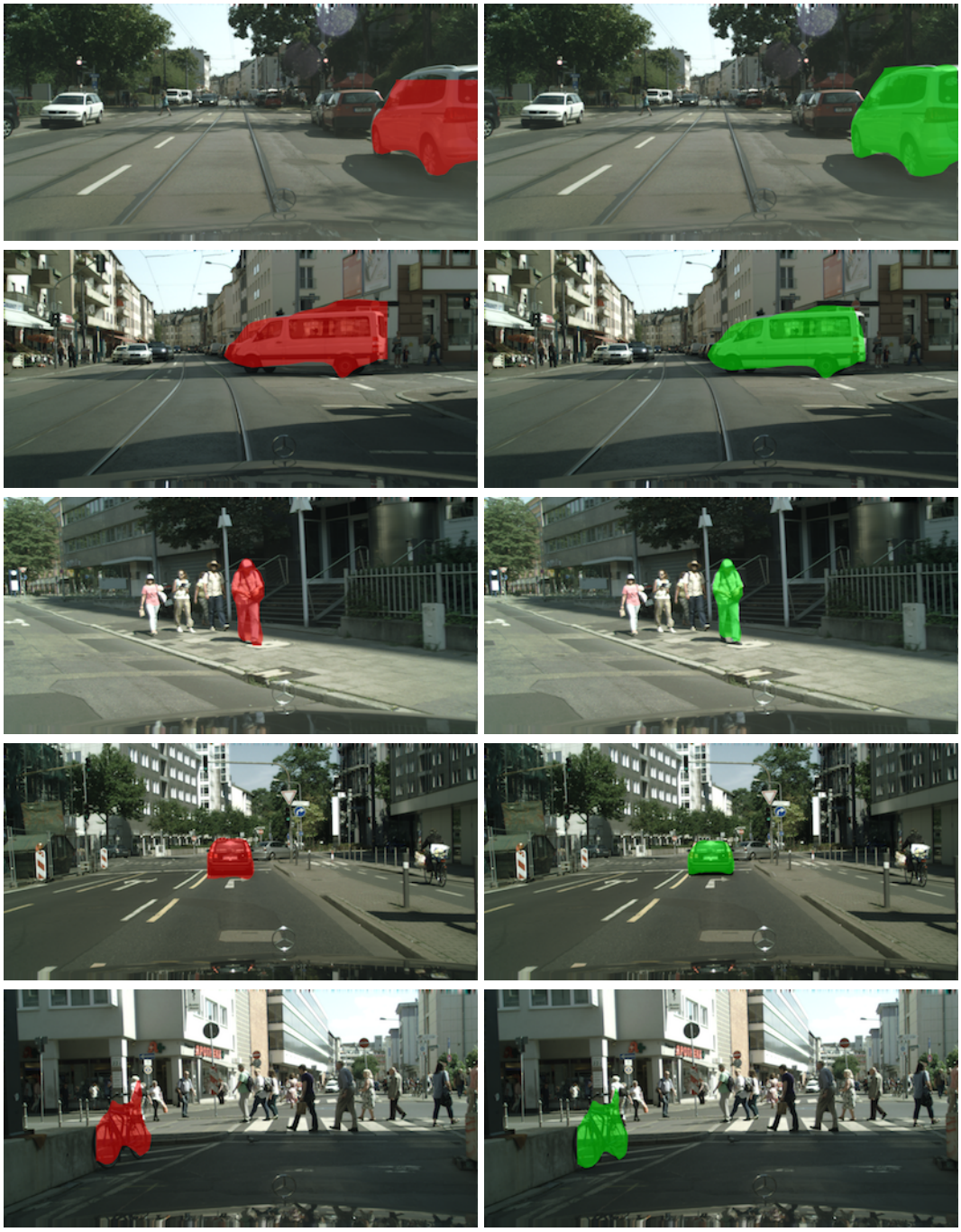


Figure 6.9: Qualitative examples for segment proposal refinement on **Cityscapes**. Red: original object mask. Green: aligned mask.

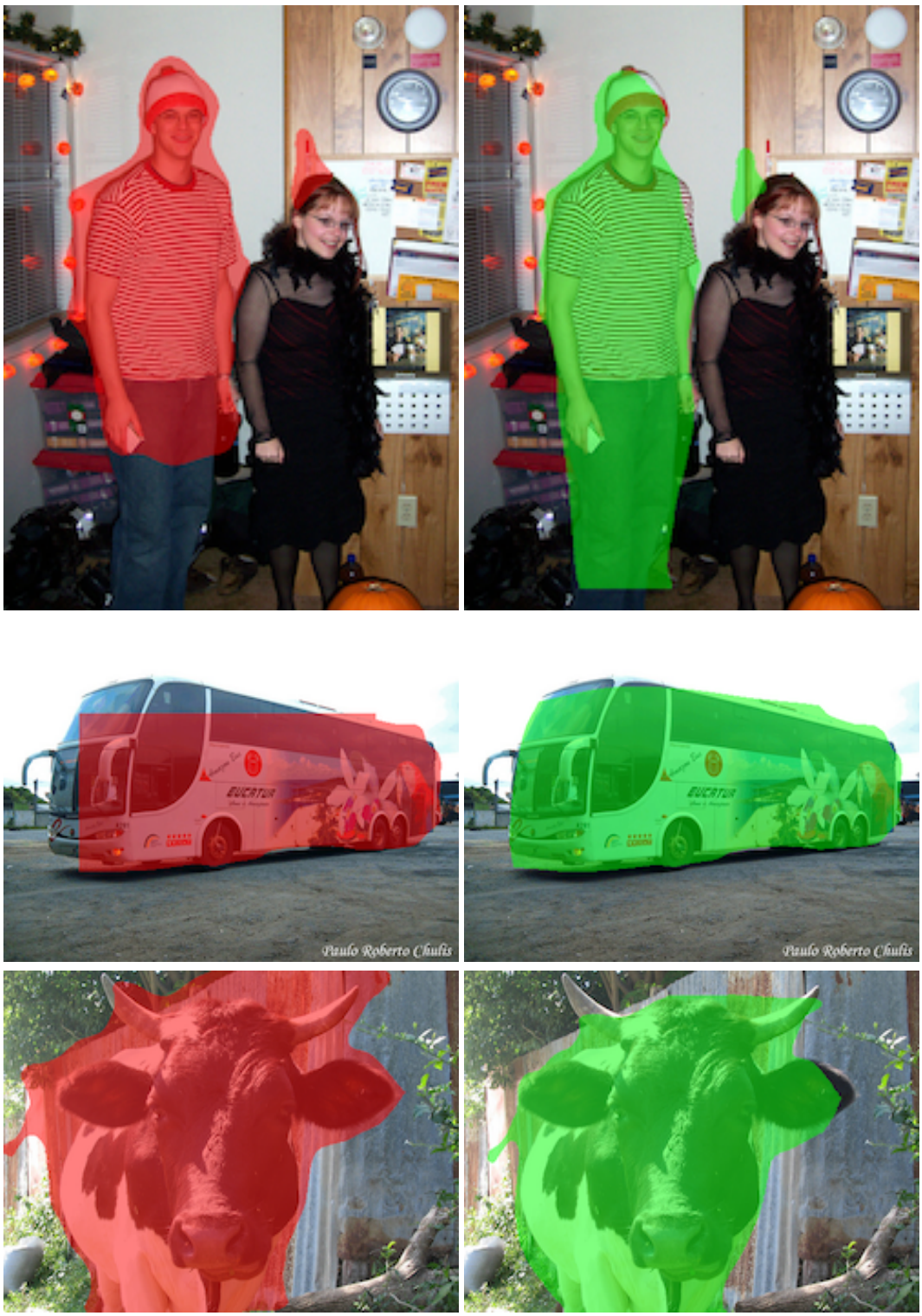


Figure 6.10: Qualitative results on **PASCAl VOC**. Red: original object mask. Green: aligned mask.

Figure 6.11: Qualitative results on **PASCAl VOC**. Red: original object mask. Green: aligned mask.

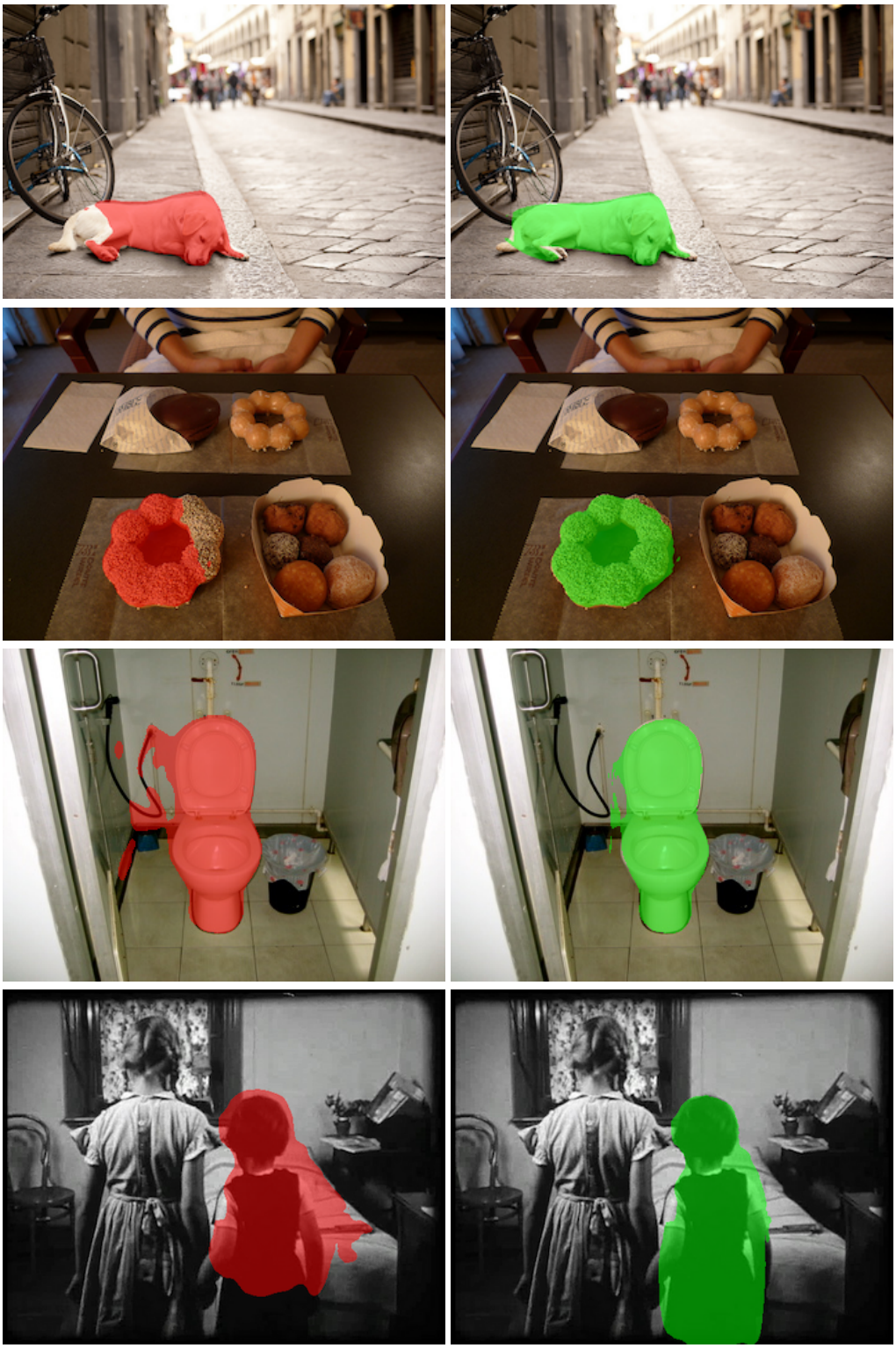


Figure 6.12: Qualitative results on **MSCOCO**. Red: original object mask. Green: aligned mask.

Chapter 7

Conclusion and Future Direction

In this thesis, we mainly investigate the problem of object proposal generation. We have developed and implemented several algorithms to generate better object proposals, especially segment proposals. This final chapter summarizes the main contributions of this thesis and closes with suggestions of possible directions for future work.

7.1 Main Contributions

Object proposal generation has become a critical step in many compute vision tasks like object detection and object instance segmentation *etc*. This thesis extends the object proposal generation to stereo images, proposes incorporating geometric information, semantic context and representation learning into proposal generation, as well as develops two transformation-based methods to refine segment proposals. In particular, we focus on three main aspects in the problem of object proposal generation: 1) generating object bounding box proposals for stereo images with geometric features and semantic context, 2) generating object segment proposals for stereo images with learning representations and learning grouping process, and 3) learning to warp object segment proposals.

We first consider the problem of generating bounding box proposals with additional geometric information and semantic context for stereo images in Chapter 3. We compute a new objectness score for each initial bounding box proposal based on three types of features, including a CNN feature, a geometric feature computed from the depth map and a semantic context feature from pixel-wise scene labelling. We train an efficient random forest classifier to predict the objectness score. To refine the location of the proposal, we also learn a set of bounding box location regressors to fine-tune the positions of the re-ranked object proposals. We evaluate our method on the KITTI dataset and achieve high recall rate with a fraction of the initial proposals, outperforming the state-of-the-art.

In Chapter 4, we move our focus to the problem of generating object segment proposal for stereo images. We propose to exploit both deep features and depth cue in segment proposal generation. For each image region, we extract a descriptor from convolutional feature maps and geometry maps to describe it, which encodes the

image region with multi-level and multi-modal information. We learn a similarity network to estimate the affinity between two adjacent regions, and based on the predicted affinity score, we sequentially merge regions from a segmentation hierarchy to produce segment proposals. We also learn a ranking network to predict the objectness score for each segment proposal. The learned representation and perceptual grouping strategy bring significant boost to the performance of segment proposal generation. Experiments on the Cityscapes dataset show that our approach achieves much better average recall than the state-of-the-art and depth cue can improve the ranking of proposals.

To generate better object segment proposals, an alternative approach is to refine an initial set of object segments. Chapter 5 presents an efficient object segment refinement method that learns spatial transforms to improve the pixel-level accuracy of the object proposals. We design a new mask pooling strategy to encode the misalignment between the segment mask and the object region. We apply the mask pooling to the hypercolumn feature maps and extract features at different levels for each segment mask. Based on the features, we design and train a deep network to predict the affine transformation parameters to warp the initial segment masks towards groundtruth object regions. We evaluate our approach on the Cityscapes and the PASCAL VOC datasets. The results demonstrate that our method can consistently achieve improvements on the IoU quality of the object segment proposals over state-of-the-art methods.

In Chapter 6, we propose a deep learning approach to address the object-mask alignment problem and apply it to the task of refining a set of segment proposals. Aligning a shape mask to object instances is a commonly used strategy in object segmentation, which can also be used in object proposal generation. We build a deep free-from deformation (FFD) network to solve this problem. Our FFD network learns a non-rigid 2D transform that warps the mask onto the target object. It consists of two modules. The first module computes multi-level features based on a dual mask feature pooling method to encode the shape information of the initial mask and the image cues around the object. The second module predicts a non-rigid transform through regression, and then applies the transform to the initial mask, based on a grid generator and a bilinear sampler, to produce the final warped object mask. Both of the modules are differentiable, making the entire network can be trained in an end-to-end fashion. We evaluate the FFD network on the task of refining a set of object segment proposals. Experiments on the challenging Cityscapes, PASCAL VOC and MSCOCO datasets show that our approach achieves the state-of-the-art performance.

7.2 Perspectives for Future Work

7.2.1 3D Object Proposals

Compared to numerous works on generating object proposals for RGB images, quite few algorithms [65, 117] have been proposed to generate 3D object proposals for

RGB-D images or point clouds. Similar to the important role of 2D object proposals playing in object detection and object instance segmentation, 3D object proposals can be expected to play a critical part in 3D visual tasks. Therefore, generating 3D object proposals for 3D scenes is a very promising research area.

One main challenge is the significantly increased search space, due to an extra dimension. In 2D space, the naive sliding window scoring approach to generating bounding box proposals already has to process millions of candidate windows. With an extra dimension, the number of 3D boxes to be scored rises sharply, so naively traversing sliding boxes is not practically feasible. Hence, a more intelligent search strategy needs to be proposed.

Additionally, learning feature representations from 3D data, like point clouds, for encoding the objectness is not trivial. Deep learning has showed great power in learning 2D image features, but adopting deep networks in 3D settings meets new difficulties. This first requires a large amount of 3D labelled data, which is expensive to obtain. Also it needs designing appropriate network components to process this form of data.

7.2.2 Generating Object Segment Proposals with Semantic Boundary Estimation

Currently, top-performing CNN-based approaches to generating segment proposals take an image-to-mask mapping method, in which an image patch is mapped into a binary mask associated with an objectness score. Such an image-to-mask mapping is often challenging to learn and the generated segment proposals tend to have bad boundary alignment. From another perspective, if we can predict a semantic boundary map for an image, on which objects are separated from the background and also object instances are divided by the boundaries, then we can obtain the segment proposals from this boundary map by simply taking those object regions.

The main difficulty would be to train such a semantic boundary estimation neural network. It is easy to predict the boundaries between the objects and the background, however, it would be probably fairly difficult to separate the object instances, especially those adjacent ones. To design such a network and choose appropriate training examples need a lot of work.

Another possible challenge might be to generate closed boundaries. The boundaries estimated by the network are likely to be disconnected, which cannot form object regions. Hence, how to obtain closed boundaries or contours from those disconnected ones is a great challenge.

7.2.3 Integrating the FFD module into Object Instance Segmentation

In this thesis, we have studied using the FFD network to refine segment proposals. Another interesting direction is to directly insert the FFD module into a instance segmentation network. There are two ways to implement this idea.

One is to intergrate the FFD module into the segment proposal generation stage

of a instance segmentation network. This is similar to our work in this thesis, but the network is a complete instance segmentation system, rather than just a proposal refinement network. The function of the FFD module in such a system is like the role of the bounding box regression part in a region proposal network [10]. This helps optimize the proposal generation for the final target.

The other is to insert the FFD module into the final stage of the object instance segmentation as a post-processing step. This helps refine those initial instance segments to make them align better with the groundtruth objects.

7.2.4 Fusing Multiple Proposals

Most existing proposal methods focus on generating a moderate set of object candidates. Few work has considered the relations between those proposals inside this set. In general, the generated proposals tend to cover part of the objects. An interesting direction would be to integrate multiple incomplete object candidates into a better complete one.

A simple method is to weighed average multiple proposals who have certain overlaps. The weights can be their objectness scores. But this method might be insufficient as it does not fully consider the relations of those overlapped proposals. A more sophisticated algorithm needs to be designed to intelligently combine the proposals.

Bibliography

1. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. (cited on pages xiii, xiv, 1, 37, 38, and 87)

2. Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. (cited on pages xiii, 2, 15, and 16)

3. Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. *arXiv preprint arXiv:1512.04412*, 2015. (cited on pages xiii, 2, 4, 5, 18, 25, 26, 30, 51, 65, and 88)

4. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010. (cited on pages xiv, 2, 5, 16, 19, 20, and 51)

5. J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015. (cited on pages xiv, 5, 18, 21, 22, 41, 51, 52, 53, 57, 60, 65, 68, 72, and 88)

6. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. (cited on pages xiv, 24, 25, 29, and 43)

7. Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. (cited on pages xiv, 26, 27, 54, and 66)

8. Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. (cited on pages xiv, 3, 4, 17, 25, 26, 27, 29, 31, 41, 65, 69, 84, 87, and 93)

9. Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015. (cited on pages xiv, 4, 18, 27, 28, 54, 65, 70, and 84)

10. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural*

Information Processing Systems, pages 91–99, 2015. (cited on pages xiv, 2, 4, 17, 25, 26, 29, 30, 31, 41, 65, and 104)

11. Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1981–1989, 2015. (cited on pages xiv, 5, 25, 26, 30, 39, 51, 62, 65, 68, 72, and 88)

12. Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. (cited on pages xiv, 25, 26, 30, 31, 38, 39, 65, 68, 72, and 88)

13. Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. (cited on pages xiv, 32, 33, and 86)

14. Fumihiko Ino, Kanrou Ooyama, and Kenichi Hagihara. A data distributed parallel algorithm for nonrigid image registration. *Parallel Computing*, 31(1):19–43, 2005. (cited on pages xiv and 34)

15. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013. (cited on pages xiv, 35, 36, 42, and 45)

16. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *arXiv preprint arXiv:1604.01685*, 2016. (cited on pages xiv, 36, 37, 52, 58, 59, 62, 66, 71, and 87)

17. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. (cited on pages xiv, 36, 37, 66, 71, and 87)

18. Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (cited on pages xiv, 37, and 54)

19. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. (cited on pages 2 and 15)

20. Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. (cited on pages 2, 15, 17, and 31)

21. Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000. (cited on page 2)

22. Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014. (cited on page 2)

23. Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. (cited on page 2)

24. Hans-Lukas Teuber. Physiological psychology. *Annual review of psychology*, 6(1):267–296, 1955. (cited on page 2)

25. Jeremy M Wolfe and Todd S Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature reviews neuroscience*, 5(6):495–501, 2004. (cited on page 2)

26. Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995. (cited on page 2)

27. Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987. (cited on page 2)

28. Charles T Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971. (cited on page 2)

29. O Morris, M Lee, and A Constantinides. A unified method for segmentation and edge detection using graph theory. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, volume 11, pages 2051–2054. IEEE, 1986. (cited on page 2)

30. Max Wertheimer. Laws of organization in perceptual forms. 1938. (cited on page 2)

31. Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012. (cited on pages 2, 5, 21, 22, 41, 45, 51, 53, 57, 58, and 65)

32. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015. (cited on pages 2, 4, 16, 24, 25, 26, 31, 41, 43, 45, and 60)

33. Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014*, pages 297–312. Springer, 2014. (cited on pages 4 and 18)

34. Dafei Huang, Lei Luo, Zhaoyun Chen, Mei Wen, and Chunyuan Zhang. Applying detection proposals to visual tracking for scale and aspect ratio adaptability. *International Journal of Computer Vision*, pages 1–18, 2016. (cited on page 4)

35. Yang Hua, Karteek Alahari, and Cordelia Schmid. Online object tracking with proposal selection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. (cited on page 4)

36. Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014. (cited on pages 5, 19, 41, and 51)

37. C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. (cited on pages 5, 20, 29, 41, 42, 43, 45, 51, 60, and 63)

38. Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. Deepbox: Learning objectness with convolutional networks. *arXiv preprint arXiv:1505.02146*, 2015. (cited on pages 5, 25, 29, 41, and 42)

39. Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2578–2586, 2015. (cited on pages 5, 29, and 41)

40. Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. (cited on pages 5, 21, 22, 41, 51, 52, 55, 60, and 93)

41. Philipp Krähenbühl and Vladlen Koltun. Learning to propose objects. In *CVPR*, 2015. (cited on pages 5, 22, 41, and 51)

42. Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Computer Vision–ECCV 2014*, pages 725–739. Springer, 2014. (cited on pages 5, 21, 52, and 60)

43. Haoyang Zhang, Xuming He, Fatih Porikli, and Laurent Kneip. Semantic context and depth-aware object proposal generation. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1–5. IEEE, 2016. (cited on page 13)

44. Haoyang Zhang, Xuming He, and Fatih Porikli. Learning to generate object segment proposals with multi-modal cues. In *Asian Conference on Computer Vision*, pages 121–136. Springer, 2016. (cited on page 13)

45. Haoyang Zhang, Xuming He, and Fatih Porikli. Learning spatial transforms for refining object segment proposals. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 37–46. IEEE, 2017. (cited on page 13)

46. Haoyang Zhang and Xuming He. Deep free-form deformation network for object-mask registration. In *ICCV*, 2017. (cited on page 14)

47. Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (cited on page 16)

48. Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009. (cited on page 16)

49. Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 237–244. IEEE, 2009. (cited on page 16)

50. Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (cited on page 16)

51. Yi Yang, Sam Hallman, Deva Ramanan, and Charless C Fowlkes. Layered object models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1731–1743, 2012. (cited on page 17)

52. Omkar M Parkhi, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. The truth about cats and dogs. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1427–1434. IEEE, 2011. (cited on page 17)

53. Qieyun Dai and Derek Hoiem. Learning to localize detected objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3322–3329. IEEE, 2012. (cited on page 17)

54. Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3294–3301, 2013. (cited on page 17)

55. Daniel Cremers, Florian Tischhäuser, Joachim Weickert, and Christoph Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International journal of computer vision*, 50(3):295–313, 2002. (cited on pages 17 and 81)

56. Daniel Kuettel and Vittorio Ferrari. Figure-ground segmentation by transferring window masks. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 558–565. IEEE, 2012. (cited on pages 17 and 81)

57. Daniel Kuettel, Matthieu Guillaumin, and Vittorio Ferrari. Segmentation propagation in imagenet. In *European Conference on Computer Vision*, pages 459–473. Springer, 2012. (cited on page 17)

58. Xuming He and Stephen Gould. An exemplar-based crf for multi-instance object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–303, 2014. (cited on pages 17, 65, and 81)

59. Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3001–3008, 2013. (cited on pages 17 and 81)

60. Alin Ionut Popa and Cristian Sminchisescu. Parametric image segmentation of humans with structural shape priors. In *ACCV*, 2016. (cited on pages 17 and 81)

61. Esa Rahtu, Juho Kannala, and Matthew Blaschko. Learning a category independent object detection cascade. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1052–1059. IEEE, 2011. (cited on page 19)

62. Ziming Zhang, Jonathan Warrell, and Philip HS Torr. Proposal generation for object detection using cascaded ranking svms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1497–1504. IEEE, 2011. (cited on page 19)

63. Michael Van den Bergh, Gemma Roig, Xavier Boix, Santiago Manen, and Luc Van Gool. Online video seeds for temporal window objectness. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 377–384, 2013. (cited on pages 19 and 20)

64. Xiaozhi Chen, Huimin Ma, Xiang Wang, and Zhichen Zhao. Improving object proposals with multi-thresholding straddling expansion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2587–2595, 2015. (cited on pages 19 and 20)

65. Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015. (cited on pages 20, 41, 45, 51, and 102)

66. Ian Endres and Derek Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010. (cited on pages 21 and 51)

67. Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2536–2543, 2013. (cited on page 21)

68. Pekka Rantalankila, Juho Kannala, and Esa Rahtu. Generating object segmentation proposals using global and local search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2417–2424, 2014. (cited on page 22)

69. Ahmad Humayun, Fuxin Li, and James M Rehg. Rigor: Reusing inference in graph cuts for generating object regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–343, 2014. (cited on page 22)

70. Victoria Yanulevskaya, Jasper Uijlings, and Nicu Sebe. Learning to group objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3134–3141, 2014. (cited on pages 22 and 55)

71. Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. (cited on page 22)

72. Tom Lee, Sanja Fidler, and Sven Dickinson. Learning to combine mid-level cues for object proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1680–1688, 2015. (cited on page 22)

73. Chaoyang Wang, Long Zhao, Shuang Liang, Liqing Zhang, Jinyuan Jia, and Yichen Wei. Object proposal by multi-branch hierarchical segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3873–3881, 2015. (cited on page 22)

74. Michael Bleyer, Christoph Rhemann, and Carsten Rother. Extracting 3d scene-consistent object proposals and depth from stereo images. In *Computer Vision–ECCV 2012*, pages 467–481. Springer, 2012. (cited on pages 22 and 51)

75. Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. (cited on page 23)

76. Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *Computer Vision–ECCV 2014*, pages 756–771. Springer, 2014. (cited on pages 23, 43, and 54)

77. Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. (cited on pages 23 and 24)

78. Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1997. (cited on page 23)

79. Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Robust monocular epipolar flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1862–1869, 2013. (cited on page 24)

80. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. (cited on pages 24, 25, 26, 53, 67, and 69)

81. Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015. (cited on pages 24 and 25)

82. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. (cited on pages 25, 43, and 86)

83. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. (cited on page 25)

84. Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2014. (cited on page 25)

85. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. (cited on pages 25 and 54)

86. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. (cited on page 25)

87. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. (cited on page 26)

88. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014. (cited on page 26)

89. Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2154, 2014. (cited on page 29)

90. Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. *arXiv preprint arXiv:1605.01014*, 2016. (cited on page 34)

91. Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016. (cited on page 34)

92. Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989. (cited on page 34)

93. Seungyong Lee, George Wolberg, and Sung Yong Shin. Scattered data interpolation with multilevel b-splines. *IEEE transactions on visualization and computer graphics*, 3(3):228–244, 1997. (cited on pages 34, 35, and 85)

94. Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999. (cited on pages 34, 66, 68, 71, 82, and 85)

95. Xiaolei Huang, Nikos Paragios, and Dimitris N Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1303–1318, 2006. (cited on page 34)

96. M Ersin Yumer and Niloy J Mitra. Learning semantic deformation flows with 3d convolutional networks. In *European Conference on Computer Vision*, pages 294–311. Springer, 2016. (cited on page 35)

97. Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. (cited on pages 36, 66, 71, 87, and 88)

98. Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *arXiv preprint arXiv:1502.05082*, 2015. (cited on pages 38, 41, 59, and 88)

99. Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. (cited on page 38)

100. Shuai Zheng, Victor Adrian Prisacariu, Melinos Averkiou, Ming-Ming Cheng, Niloy J Mitra, Jamie Shotton, Philip HS Torr, and Carsten Rother. Object proposals estimation in depth image using compact 3d shape manifolds. In *Pattern Recognition*, pages 196–208. Springer, 2015. (cited on page 41)

101. Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *CVIU*, 114:712–722, 2010. (cited on page 41)

102. Derek Hoiem, Alexei a. Efros, and Martial Hebert. Putting Objects in Perspective. *IJCV*, 80:3–15, 2008. (cited on page 41)

103. Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009. (cited on page 41)

104. Joseph Tighe and Svetlana Lazebnik Marc Niethammer. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. (cited on page 41)

105. Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. (cited on page 43)

106. Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. (cited on page 43)

107. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. (cited on page 43)

108. Piotr Dollár. Piotr's computer vision matlab toolbox (pmt). http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html. (cited on page 44)

109. Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(8):1558–1570, 2015. (cited on page 53)

110. Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 689–692. ACM, 2015. (cited on pages 54, 56, and 70)

111. Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. *arXiv preprint arXiv:1511.08498*, 2015. (cited on page 65)

112. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. (cited on page 70)

113. Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. *arXiv preprint arXiv:1612.05478*, 2016. (cited on page 81)

114. Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003. (cited on page 81)

115. Charles R Meyer, Jennifer L Boes, Boklye Kim, Peyton H Bland, Kenneth R Zasadny, Paul V Kison, Kenneth Koral, Kirk A Frey, and Richard L Wahl. Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations. *Medical image analysis*, 1(3):195–206, 1997. (cited on page 81)

116. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. (cited on page 86)

117. Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. (cited on page 102)